ANDRÉA ROSELI VANÇAN RUSSO HORIMOTO

**ESTIMATIVA DO VALOR DA TAXA DE PENETRÂNCIA EM
DOENÇAS AUTOSSÔMICAS DOMINANTES: ESTUDO TEÓRICO
DE MODELOS E DESENVOLVIMENTO DE UM
PROGRAMA COMPUTACIONAL**

São Paulo
2009

ANDRÉA ROSELI VANÇAN RUSSO HORIMOTO

**ESTIMATIVA DO VALOR DA TAXA DE PENETRÂNCIA EM
DOENÇAS AUTOSSÔMICAS DOMINANTES: ESTUDO TEÓRICO
DE MODELOS E DESENVOLVIMENTO DE UM
PROGRAMA COMPUTACIONAL**

Tese apresentada ao Instituto de
Biociências da Universidade de São
Paulo, como requisito para obtenção
do título de Doutor em Ciências, na
Área de Biologia/Genética.

Orientador: Prof. Dr. Paulo A. Otto

São Paulo
2009

Ficha Catalográfica

Horimoto, A.R.V.R.

Estimativa do valor da taxa de penetrância em doenças autossômicas dominantes: estudo teórico de modelos e desenvolvimento de um programa computacional.

265p.

Tese (Doutorado) - Instituto de Biociências da Universidade de São Paulo. Departamento de Genética e Biologia Evolutiva.

1. taxa de penetrância, 2. modelos matemáticos, 3. programa computacional, 4. método de Bayes, 5. método de máxima verossimilhança. I. Universidade de São Paulo. Instituto de Biociências. Departamento de Genética e Biologia Evolutiva.

Comissão Julgadora:

Prof(a).Dr(a).

Prof(a).Dr(a).

Prof(a).Dr(a).

Prof(a).Dr(a).

Prof. Dr. Paulo A. Otto
Orientador

Ao meu marido Edson e aos meus
filhos Felipe e Danilo, pelo
amor e dedicação.

À minha mãe Doraci, pelas palavras
de incentivo, apoio e amor.

AGRADECIMENTOS

Ao Prof. Dr. Paulo A. Otto, pela orientação, apoio e amizade ao longo dos anos de convivência acadêmica.

Ao Dr. Peter L. Pearson, pelas sugestões e importante contribuição.

Ao Prof. Dr. Sérgio Russo Mاتيoli, pelas sugestões.

Ao graduando Márcio Teruo Onodera, pelas sugestões e desenvolvimento paralelo da versão web do programa PenCalc.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro a essa pesquisa.

Aos amigos do Laboratório de Genética Humana, em especial à Eliete e à Carola, pelas sugestões e amizade.

Ao meu marido Edson e aos meus filhos Felipe e Danilo, pela compreensão, paciência e apoio incondicional a esse trabalho.

Aos meus pais, que se dedicaram intensamente à realização dos meus sonhos e demais familiares, pela ajuda e dedicação.

À minha amiga Priscilla Ueno, pelo apoio e incentivo nos momentos mais difíceis.

Aos professores e funcionários do Instituto de Biociências da Universidade de São Paulo.

E a todos que direta ou indiretamente contribuíram para a realização desse trabalho.

ÍNDICE

I) INTRODUÇÃO.....	1
1) Métodos de estimativa da taxa de penetrância.....	5
1.1) Estimativa da taxa de penetrância empregando dados familiares.....	5
1.2) Estimativa da taxa de penetrância empregando dados de gêmeos.....	9
1.3) Estimativa da taxa de penetrância utilizando informações genealógicas	14
2) Bases genéticas da penetrância incompleta	16
2.1) Pré-mutação.....	16
2.2) Ação epistática de genes modificadores.....	19
2.3) Influência do alelo em trans.....	20
2.4) <i>Imprinting</i> genômico.....	20
2.5) Genes supressores de tumor.....	21
3) Síndrome da ectrodactilia associada à hemimelia tibial.....	22
II) OBJETIVOS.....	25
III) MÉTODOS.....	27
1) Modelo	27
2) Estimativa da taxa de penetrância.....	27
2.1) Estruturas da genealogia.....	27
2.1.1) Indivíduo gerador da genealogia.....	28
2.1.2) Indivíduo afetado.....	28
2.1.3) Indivíduo normal, heterozigoto certo.....	29
2.1.4) Indivíduo normal, sem filhos, descendente de um heterozigoto certo.....	29
2.1.5) Árvores de indivíduos normais.....	29
2.1.6) Indivíduo resultante de união consanguínea.....	30
2.1.7) Árvores com uniões consanguíneas.....	30

2.1.8) Gêmeos.....	34
2.2) Filtragem da genealogia.....	38
2.3) Função de verossimilhança.....	38
2.4) Método de máxima verossimilhança.....	39
3) Intervalo exato de confiança a 95%.....	41
4) Probabilidade de heterozigose e cálculo de risco.....	42
4.1) Árvores de indivíduos normais com uma geração.....	42
4.2) Árvores de indivíduos normais com duas gerações....	43
5) Teste de razão de verossimilhanças.....	44
6) Alguns resultados gerais de interesse teórico.....	45
6.1) Fórmulas gerais para árvores de indivíduos normais com ramificação regular.....	45
6.2) Árvores normais com uniões consanguíneas.....	51
IV) EXEMPLOS DE APLICAÇÃO DO MÉTODO.....	60
1) Estudo de modelos com taxa de penetrância constante e variável.....	60
1.1) Modelo com taxa de penetrância constante (modelo I).....	60
1.2) Modelos com taxa de penetrância variável.....	66
1.2.1) Modelo com um valor de K por geração (modelo IV)	66
1.2.2) Modelo com dois valores de K (modelo II).....	73
1.2.3) Teste de razão de verossimilhanças (teste G).....	78
2) Análise de genealogias descritas na literatura.....	79
2.1) Genealogia 1 (Richieri-Costa et al., 1987).....	80
2.2) Genealogia 2 (Richieri-Costa et al., 1987).....	82
2.3) Genealogia 4 (Richieri-Costa et al., 1987).....	83
2.4) Genealogia 5 (Richieri-Costa et al., 1987).....	85
2.5) Genealogia 6 (Richieri-Costa et al., 1987).....	86
2.6) Genealogia 7 (Hoyme et al., 1987).....	88

2.7) Genealogia 8 (Hoyme et al., 1987).....	89
2.8) Genealogia 9 (Hoyme et al., 1987).....	93
2.9) Genealogia 10 (Hoyme et al., 1987).....	94
2.10) Genealogia 11 (Spranger & Schapera, 1988).....	95
2.11) Genealogia 12 (Majewski et al., 1985).....	97
2.12) Genealogia 13 (Majewski et al., 1985).....	103
2.13) Genealogia 14 (Majewski et al., 1985).....	104
2.14) Genealogia 15 (Majewski et al., 1985).....	106
2.15) Genealogia 16 (Majewski et al., 1985).....	110
2.16) Genealogia 17 (Witters et al., 2001).....	111
2.17) Genealogia 18 (Sener et al., 1989).....	112
2.18) Genealogia 19 (Marinoni et al., 1994).....	116
2.19) Genealogia 20 (Naveed et al., 2006).....	125
2.20) Genealogia 21 (Majewski et al., 1996).....	132
3) Meta-análise.....	136
3.1) Modelo com taxa de penetrância constante.....	136
3.2) Modelo com taxa de penetrância variável.....	137
V) PROGRAMAS COMPUTACIONAIS.....	139
1) PenCalc v. 1.0: manual de instruções.....	139
2) PenCalc Web v. 1.0: uma versão do programa para internet.....	161
VI) RESUMO.....	168
VII) ABSTRACT.....	170
VIII) REFERÊNCIAS BIBLIOGRÁFICAS.....	172
IX) ANEXO I.....	180
X) ANEXO II.....	193

I) INTRODUÇÃO

A taxa de penetrância (**K**) é a probabilidade condicional p_{ij} de manifestação fenotípica de um genótipo $a_i a_j$, qualquer. Como toda probabilidade, a taxa de penetrância pode assumir qualquer valor no intervalo entre zero e um; quando a penetrância assume o valor 1, é dita completa; quando assume qualquer valor maior que zero e menor que um, é dita incompleta. Inversamente, a não penetrância de um gene, cuja probabilidade é definida como $1 - p_{ij}$, é a falta de manifestação fenotípica devido exclusiva ou predominantemente a fatores genéticos aleatórios associados ao genótipo $a_i a_j$, como mutações ou ação epistática.

O conceito de penetrância, originalmente descrito por Vogt (1926), foi posteriormente modificado ou generalizado por vários autores (Berglin, 1955, 1957; Trankell, 1956; Sang, 1963; Wilson, 1971; Suarez et al., 1976; Cannings et al., 1980; Matsunaga, 1980).

No caso generalizado, a taxa de penetrância pode ser expressa pela matriz de probabilidades transicionais p_{ij} (Rogatko, 1983):

$$\begin{array}{ccc} & f_1 & f_2 & f_3 \\ \begin{array}{c} a_1 a_1 \\ a_1 a_2 \\ a_2 a_2 \end{array} & \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix} & , \end{array}$$

onde $a_1 a_1$, $a_1 a_2$ e $a_2 a_2$ são os genótipos determinados pelas combinações possíveis dos alelos a_1 e a_2 e f_1 , f_2 e f_3 representam as classes fenotípicas. Desta forma, a probabilidade condicional $p(f_k | a_i a_j)$ de um indivíduo apresentar o fenótipo f_k , dado que seu genótipo é $a_i a_j$, é a taxa de penetrância p_{ij} . Como p_{ij} é uma matriz de transição, as suas linhas somam a unidade:

$$\sum_{j=1}^3 p_{ij} = 1.$$

Nos casos particulares de dominância e codominância com penetrância completa, a matriz transicional $\mathbf{p}_{ij}$ se reduz respectivamente (Rogatko, 1983) a

$$\begin{array}{c} \mathbf{f}_1 \quad \mathbf{f}_2 \quad \mathbf{f}_3 \\ \mathbf{a}_1\mathbf{a}_1 \quad \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ \mathbf{a}_1\mathbf{a}_2 \quad \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ \mathbf{a}_2\mathbf{a}_2 \quad \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \end{array}$$

e

$$\begin{array}{c} \mathbf{f}_1 \quad \mathbf{f}_2 \quad \mathbf{f}_3 \\ \mathbf{a}_1\mathbf{a}_1 \quad \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \\ \mathbf{a}_1\mathbf{a}_2 \quad \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \\ \mathbf{a}_2\mathbf{a}_2 \quad \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \end{array}$$

Os valores de penetrância $\mathbf{p}_{ij}$, em todos os casos possíveis de dominância, codominância e recessividade com penetrância incompleta, podem ser reduzidos ao parâmetro $\mathbf{p}_{ij} = \mathbf{K}$, o qual foi utilizado para definir cinco modelos distintos de penetrância, como apresentado na Tabela 1 (Otto et al., 1994). As linhas das matrizes correspondem às classes genotípicas ($\mathbf{a}_1\mathbf{a}_1$, $\mathbf{a}_1\mathbf{a}_2$ e $\mathbf{a}_2\mathbf{a}_2$, respectivamente) e as colunas, às classes fenotípicas (**dominante** e **recessiva**, respectivamente).

Tabela 1 - Matrizes de transição aplicáveis a modelos dominantes e recessivos com penetrância (K) incompleta (Otto et al., 1994).

MODELO		MATRIZ DE TRANSIÇÃO
DOMINANTE	I	$\begin{bmatrix} 1 - (1 - K)^2 & (1 - K)^2 \\ K & 1 - K \\ 0 & 1 \end{bmatrix}$
	II	$\begin{bmatrix} 1 & 0 \\ K & 1 - K \\ 0 & 1 \end{bmatrix}$
	III	$\begin{bmatrix} K & 1 - K \\ K & 1 - K \\ 0 & 1 \end{bmatrix}$
RECESSIVO	IV	$\begin{bmatrix} 1 & 0 \\ 1 & 0 \\ K & 1 - K \end{bmatrix}$
	V	$\begin{bmatrix} 1 & 0 \\ K & 1 - K \\ K^2 & 1 - K^2 \end{bmatrix}$

O modelo I considera os efeitos dos dois alelos independentes no homozigoto a_1a_1 , de forma que a taxa de penetrância é dada por $2K(1-K) + K^2 = 1 - (1-K)^2$. Os modelos II e III são modelos padrão em genética humana. Os modelos IV e V consideram o genótipo recessivo incompletamente penetrante.

Alguns modelos alternativos de penetrância incompleta para genes dominantes e codominantes são descritos na Tabela 2 (Rogatko, 1983). As linhas das matrizes correspondem às classes genotípicas e as colunas, às classes fenotípicas definidas anteriormente.

Tabela 2 - Matrizes de transição aplicáveis a alguns modelos dominantes (III e IV) e codominantes (VI, VII e VIII) com penetrância (K) incompleta (adaptada de Rogatko, 1983).

MODELO	MATRIZ DE TRANSIÇÃO
III	$\begin{bmatrix} 1 & 0 & 0 \\ K & 0 & 1-K \\ 0 & 0 & 1 \end{bmatrix}$
IV	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1-K & 0 & K \end{bmatrix}$
VI	$\begin{bmatrix} K & 1-K & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
VII	$\begin{bmatrix} K & 0 & 1-K \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
VIII	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & K & 1-K \\ 0 & 0 & 1 \end{bmatrix}$

Os conjuntos de probabilidades condicionais indicados nas Tabelas 1 e 2 são válidos para características normais polimórficas (frequentes) na população, podendo apenas o modelo II (Tabela 1) ser também aplicado a características patológicas (raras).

No caso de doenças monogênicas autossômicas dominantes, o fenótipo associado aos indivíduos homozigotos a_1a_1 geralmente não é conhecido, de forma que o valor 1 atribuído a esses indivíduos na matriz de transição (modelo II da Tabela 1)

pode ser substituído por zero. Duas hipóteses explicam a inexistência de indivíduos a_1a_1 : ou o gene a_1 é embriologicamente letal em homozigose ou apresenta uma frequência populacional tão baixa que, exceto sob uniões consanguíneas ou preferenciais, é praticamente impossível o aparecimento de indivíduos homozigotos.

Desta forma, uma fração K de indivíduos heterozigotos manifesta fenotipicamente a condição (penetrantes), enquanto uma fração $1-K$ de heterozigotos (não penetrantes) e todos os homozigotos a_2a_2 são normais, constituindo uma classe de indivíduos fenotipicamente indistinguíveis.

Sob hipótese de pan-mixia, as frequências populacionais de indivíduos com fenótipo dominante são as seguintes, em relação aos modelos da Tabela 1:

Tabela 3 - Frequências populacionais de indivíduos com fenótipo dominante sob os modelos de penetrância I a V (Tabela 1).

MODELO	P (dom, a_1a_1)	P (dom, a_1a_2)	P (dom, a_2a_2)	P (dom)
I	$p^2K(2-K)$	$2pqK$	0	$pK(2-pK)$
II	p^2	$2pqK$	0	$p(p+2qK)$
III	p^2K	$2pqK$	0	$pK(1+q)$
IV	p^2	$2pq$	q^2K	$1-q^2(1-K)$
V	p^2	$2pqK$	q^2K^2	$(p+qK)^2$

Essas quantidades são utilizadas para estimar o valor da taxa de penetrância através de métodos que empregam dados familiares ou pares gêmeares, como descrito a seguir.

1) MÉTODOS DE ESTIMATIVA DA TAXA DE PENETRÂNCIA

A taxa de penetrância pode ser estimada por regra de três baseada em casos verificados de penetrância e não penetrância (Stern, 1973); pela análise de dados populacionais ou familiares (Trankell, 1955; Elandt-Johnson, 1970; Suarez et al., 1976, 1977; Otto et al., 1994), que requerem a aplicação de métodos mais elaborados; empregando dados de gêmeos mono ou dizigóticos (Praxedes, 1997; Praxedes & Otto, 2000) ou pela análise de dados genealógicos (Rogatko, 1983; Rogatko et al., 1986).

Com exceção dos métodos descritos por Stern (1973) e Rogatko et al. (1986), nenhum dos demais se aplica à análise de doenças genéticas, a qual exige uma metodologia de estimação do valor da taxa de penetrância ajustada a genes com frequência baixa na população.

1.1) ESTIMATIVA DA TAXA DE PENETRÂNCIA EMPREGANDO DADOS FAMILIAIS

Otto et al. (1994) adaptaram os métodos de análise de segregação aplicados a genes polimórficos na população, originalmente desenvolvidos por Snyder (1932, 1934) e Fisher (Taylor & Prior, 1939), de forma a incorporar o parâmetro adicional da taxa de penetrância nos modelos correspondentes.

As proporções de indivíduos recessivos (**R**) na progênie de casais dominante × dominante (**DD**), dominante × recessivo (**DR**) e recessivo × recessivo (**RR**) são dadas por:

$$P_1 = P(R|DD) = \sum DD_i R_i / \sum DD_i, \quad P(DD) = \sum DD_i$$

$$P_2 = P(R|DR) = \sum DR_i R_i / \sum DR_i, \quad P(DR) = \sum DR_i$$

$$P_3 = P(R|RR) = \sum RR_i R_i / \sum RR_i, \quad P(RR) = \sum RR_i$$

Na Tabela 4, mostramos as frequências populacionais esperadas de todos os cruzamentos fenotípicos possíveis e de suas proles recessivas, considerando-se as taxas genéricas de

penetrância K_1 , K_2 e K_3 , respectivamente, para os genótipos a_2a_2 , a_1a_2 e a_1a_1 .

Tabela 4 - Frequências esperadas, sob regime de pan-mixia, de casais dom × dom, dom × rec e rec × rec e respectivas proles recessivas, em função da frequência gênica (p e q) e da taxa de penetrância (K_j).

CASAI	DOM × DOM	DOM × REC	REC × REC	PROGÊNIE RECESSIVA
$a_1a_1 \times a_1a_1$	$DD_1 = p^4K_3^2$	$DR_1 = 2p^4K_3(1-K_3)$	$RR_1 = p^4(1-K_3)^2$	$R_1 = 1-K_3$
$a_1a_1 \times a_1a_2$	$DD_2 = 4p^3qK_2K_3$	$DR_2 = 4p^3qK_2(1-K_3) + 4p^3qK_3(1-K_2)$	$RR_2 = 4p^3q(1-K_2) \cdot (1-K_3)$	$R_2 = [(1-K_2) + (1-K_3)]/2$
$a_1a_1 \times a_2a_2$	$DD_3 = 2p^2q^2K_1K_3$	$DR_3 = 2p^2q^2K_1(1-K_3) + 2p^2q^2K_3(1-K_1)$	$RR_3 = 2p^2q^2(1-K_1) \cdot (1-K_3)$	$R_3 = 1-K_2$
$a_1a_2 \times a_1a_2$	$DD_4 = 4p^2q^2K_2^2$	$DR_4 = 8p^2q^2K_2 \cdot (1-K_2)$	$RR_4 = 4p^2q^2(1-K_2)^2$	$R_4 = [(1-K_1) + 2(1-K_2) + (1-K_3)]/4$
$a_1a_2 \times a_2a_2$	$DD_5 = 4pq^3K_1K_2$	$DR_5 = 4pq^3K_1 \cdot (1-K_2) + 4pq^3K_2(1-K_1)$	$RR_5 = 4pq^3 \cdot (1-K_1)(1-K_2)$	$R_5 = [(1-K_1) + (1-K_2)]/2$
$a_2a_2 \times a_2a_2$	$DD_6 = q^4K_1^2$	$DR_6 = 2q^4K_1(1-K_1)$	$RR_6 = q^4(1-K_1)^2$	$R_6 = 1-K_1$
	ΣDD_i	ΣDR_i	ΣRR_i	

Essas quantidades são utilizadas para compor as proporções P_1 , P_2 e P_3 definidas anteriormente, substituindo-se, por exemplo, as taxas genéricas K_1 , K_2 e K_3 pelos valores específicos da taxa de penetrância dos modelos I a V apresentados na Tabela 1 (Otto et al., 1994; Horimoto & Otto, 2008):

Modelo I:

$$P_1 = P(R|DD) = (2-K)^2(1-pK)^2/(2-pK)^2$$

$$P_2 = P(R|DR) = (2-K)[1-pK(2-K)]/(2-pK)$$

$$P_3 = P(R|RR) = [1-pK(1-K)/(1-pK)]^2$$

Modelo II:

$$P_1 = P(R|DD) = qK[2p(1-K) + qK(3-2K)]/(p+2qK)^2$$

$$P_2 = P(R|DR) = \{p(1-K)[1-pK+qK(1-K)] + qK(2-K)(1-pK)\} / \{(p+2qK)[2p(1-K)+q]\}$$

$$P_3 = P(R|RR) = \{p(1-K)[p(1-K)(3-2K)+2q(2-K)] + q^2\} / [2p(1-K)+q]^2$$

Modelo III:

$$P_1 = P(R|DD) = 1-K + q^2K/(1+q)^2$$

$$P_2 = P(R|DR) = [p(1-K)^2(1+q)^2 + pq^2(1-K^2) + q^3(2-K)] / \{(1+q)[1-pK(1+q)]\}$$

$$P_3 = P(R|RR) = \{(1-pK)^4 + pK(1-K)[p^3(1-K)^2 - 3pq^2(1-K) - 2q^3]\} / [1-pK(1+q)]^2$$

Modelo IV:

$$P_1 = P(R|DD) = (1-K)\{q(p+qK)/[1-q^2(1-K)]\}^2$$

$$P_2 = P(R|DR) = (1-K)q(p+qK)/[1-q^2(1-K)]$$

$$P_3 = P(R|RR) = 1-K$$

Modelo V:

$$P_1 = P(R|DD) = qK(1-K)[2-q(1-K)(2+K)]/[1-q(1-K)]^2$$

$$P_2 = P(R|DR) = (1-K)[(1-q)^2 + (1-q)^2qK(4+K) + (1-q)q^2K(2+K)(1+2K) + q^3K^2(1+K)^2] / \{[1-q(1-K)]^2[2-q(1-K)]\}$$

$$P_3 = P(R|RR) = (1-K)[(1-q)^2(3+K) + 2q(1-q)(1+K)(2+K) + q^2(1+K)^3]/[2-q(1-K)]^2$$

Da análise de uma amostra de $n_1 + n_2 + n_3$ casais com $N_1 + N_2 + N_3$ filhos, são observadas as seguintes quantidades:

Pais	Filhos		
	Dom	Rec	Total
dom x dom: n_1	n_4	n_5	N_1
dom x rec: n_2	n_6	n_7	N_2
rec x rec: n_3	n_8	n_9	N_3
Total	N_4	N_5	N

Sob a hipótese de herança monogênica dominante com penetrância incompleta, o logaritmo da função de verossimilhança é dado pela expressão:

$$\begin{aligned}
L &= n_1 \cdot \log[P(DD)] + n_2 \cdot \log[P(DR)] + n_3 \cdot \log[P(RR)] \\
&+ n_4 \cdot \log[P(D|DD)] + n_5 \cdot \log[P(R|DD)] + n_6 \cdot \log[P(D|DR)] \\
&+ n_7 \cdot \log[P(R|DR)] + n_8 \cdot \log[P(D|RR)] + n_9 \cdot \log[P(R|RR)] \\
&= (2n_1+n_2) \cdot \log[P(D)] + (n_2+2n_3) \cdot \log[P(R)] + n_4 \cdot \log(Q_1) \\
&+ n_5 \cdot \log(P_1) + n_6 \cdot \log(Q_2) + n_7 \cdot \log(P_2) + n_8 \cdot \log(Q_3) + n_9 \cdot \log(P_3),
\end{aligned}$$

onde $Q_i = 1-P_i$ ($i = 1, 2, 3$).

A estimativa de máxima verossimilhança do valor da taxa de penetrância K é obtida através da solução da equação $dL/dK = 0$ [ou, mais convenientemente, $dL/dK = 0$, onde $L = \log(P)$].

A aderência dos valores observados aos esperados pode ser avaliada aplicando-se os seguintes testes:

1) na progênie de casais dominante $\times$ dominante:

$$\begin{aligned}
\chi^2_{(1 \text{ g.l.})} &= (n_4 - N_1 Q_1)^2 / N_1 Q_1 + (n_5 - N_1 P_1)^2 / N_1 P_1 \\
&= n_4^2 / N_1 Q_1 + n_5^2 / N_1 P_1 - N_1
\end{aligned}$$

2) na progênie de casais dominante $\times$ recessivo:

$$\begin{aligned}
\chi^2_{(1 \text{ g.l.})} &= (n_6 - N_2 Q_2)^2 / N_2 Q_2 + (n_7 - N_2 P_2)^2 / N_2 P_2 \\
&= n_6^2 / N_2 Q_2 + n_7^2 / N_2 P_2 - N_2
\end{aligned}$$

3) na progênie de casais recessivo $\times$ recessivo:

$$\begin{aligned}
\chi^2_{(1 \text{ g.l.})} &= (n_8 - N_3 Q_3)^2 / N_3 Q_3 + (n_9 - N_3 P_3)^2 / N_3 P_3 \\
&= n_8^2 / N_3 Q_3 + n_9^2 / N_3 P_3 - N_3
\end{aligned}$$

4) teste de heterogeneidade:

$$\begin{aligned}
\chi^2 &= \text{soma dos valores de } \chi^2 \text{ obtidos em cada teste (3 g.l.)} \\
&- \chi^2 \text{ das somas (1 g.l.)} \\
&= n_4^2 / N_1 Q_1 + n_5^2 / N_1 P_1 + n_6^2 / N_2 Q_2 \\
&+ n_7^2 / N_2 P_2 + n_8^2 / N_3 Q_3 + n_9^2 / N_3 P_3 \\
&- N_4^2 / (N_1 Q_1 + N_2 Q_2 + N_3 Q_3) - N_5^2 / (N_1 P_1 + N_2 P_2 + N_3 P_3)
\end{aligned}$$

O programa **TONGMOD1.BAS (ANEXO I)**, desenvolve todos os cálculos necessários para estimar os valores de p e K ,

considerando o modelo I (Tabela 1). Utilizando como exemplo os dados da amostra total estudada por Otto et al. (1994) em relação à característica capacidade de enrolar a língua, o programa apresenta a seguinte saída de dados:

```
N(DOM x DOM CROS.) = ? 102
N(DOM x REC CROS.) = ? 102
N(REC x REC CROS.) = ? 22
N(DOM | DOM x DOM) = ? 241
N(REC | DOM x DOM) = ? 39
N(DOM | DOM x REC) = ? 198
N(REC | DOM x REC) = ? 98
N(DOM | REC x REC) = ? 24
N(REC | REC x REC) = ? 43
INITIAL GUESSES (P,K) = ? 0.5,0.8
```

MODEL I : $PEN(RR) = 1 - (1 - K)^2$, $PEN(Rr) = K$

p	s.e. (p)	K	s.e. (K)	dL/dp	dL/dK
0.50000	0.02627	0.80000	0.03072	175.32706	103.12791
0.56763	0.02932	0.80659	0.03395	-0.26061	-7.06936
0.57254	0.03011	0.79863	0.03493	0.10607	-0.12582
0.57274	0.03014	0.79839	0.03496	-0.00000	-0.00026
0.57274	0.03014	0.79839	0.03496	-0.00013	-0.00015

RESULTS OF GOODNESS-OF-FIT TESTS :

- a) off. of DxD parents : chi-square (1 d.f.) = 2.963
- b) off. of DxR parents : chi-square (1 d.f.) = 0.511
- c) off. of RxR parents : chi-square (1 d.f.) = 0.701
- d) 'total' chi-square : chi-square (3 d.f.) = 4.175
- e) 'pooled' chi-square : chi-square (1 d.f.) = 2.925
- f) heterogeneity c.s. : chi-square (2 d.f.) = 1.250

1.2) ESTIMATIVA DA TAXA DE PENETRÂNCIA EMPREGANDO DADOS DE GÊMEOS

O valor da taxa de penetrância também pode ser estimado a partir de dados de pares gemelares monozigóticos e dizigóticos (Praxedes, 1997; Praxedes & Otto, 2000) e de pares de irmãos (ou teoricamente de qualquer conjunto de pares de indivíduos com algum parentesco).

Detalhamos abaixo, o método proposto por Praxedes & Otto (2000), aplicado ao modelo I (Tabela 1), cuja matriz de probabilidades transicionais é dada por:

$$\begin{matrix} & \text{dom} & \text{rec} \\ \begin{matrix} a_1a_1 \\ a_1a_2 \\ a_2a_2 \end{matrix} & \begin{bmatrix} 1 - (1-K)^2 & (1-K)^2 \\ K & 1-K \\ 0 & 1 \end{bmatrix} \end{matrix}.$$

A Tabela 5 apresenta as frequências populacionais esperadas de pares de gêmeos monozigóticos e dizigóticos, sob hipótese de pan-mixia, considerando-se as combinações genotípicas (a_ia_j) e fenotípicas [dominante (**dom**) ou recessivo (**rec**)] possíveis.

Tabela 5 - Frequências populacionais esperadas de pares de gêmeos monozigóticos (MZ) e dizigóticos (DZ).

GENÓTIPO	TIPO	FENÓTIPO			TOTAL
		DOM-DOM	DOM-REC	REC-REC	
$a_1a_1 - a_1a_1$	MZ	$p^2 [1 - (1-K)^2]^2$	$2p^2 [1 - (1-K)^2] (1-K)^2$	$p^2 [(1-K)^2]^2$	p^2
	DZ	$\frac{p^2 (1+p)^2}{[1 - (1-K)^2]^2 / 4}$	$\frac{p^2 (1+p)^2}{[1 - (1-K)^2] (1-K)^2 / 2}$	$\frac{p^2 (1+p)^2}{(1-K)^4 / 4}$	$p^2 (1+p)^2 / 4$
$a_1a_1 - a_1a_2$	DZ	$\frac{p^2 q K (1+p)}{[1 - (1-K)^2]}$	$\frac{p^2 q (1+p) \{ [1 - (1-K)^2] \cdot (1-K) + (1-K)^2 K \}}{[1 - (1-K)^2]}$	$p^2 q (1+p) (1-K)^3$	$p^2 q (1+p)$
$a_1a_1 - a_2a_2$	DZ	0	$p^2 q^2 [1 - (1-K)^2] / 2$	$p^2 q^2 (1-K)^2 / 2$	$p^2 q^2 / 2$
$a_1a_2 - a_1a_2$	MZ	$2pqK^2$	$4pqK(1-K)$	$2pq(1-K)^2$	$2pq$
	DZ	$pqK^2(1+pq)$	$2pqK(1+pq)(1-K)$	$pq(1+pq)(1-K)^2$	$pq(1+pq)$
$a_1a_2 - a_2a_2$	DZ	0	$pq^2K(1+q)$	$pq^2(1+q)(1-K)$	$pq^2(1+q)$
$a_2a_2 - a_2a_2$	MZ	0	0	q^2	q^2
	DZ	0	0	$q^2(1+q)^2 / 4$	$q^2(1+q)^2 / 4$

Apresentamos na Tabela 6 as frequências esperadas dos possíveis pares de gêmeos, sendo $\{n_1, n_2, n_3\}$ e $\{n_4, n_5, n_6\}$ os números observados de pares gemelares monozigóticos e

dizigóticos, respectivamente, dominante-dominante, dominante-recessivo e recessivo-recessivo e $n_1 + n_2 + n_3 = N_{MZ}$ e $n_4 + n_5 + n_6 = N_{DZ}$ os números totais de pares de gêmeos monozigóticos e dizigóticos amostrados.

Tabela 6 - Frequências esperadas de pares de gêmeos monozigóticos (MZ) e dizigóticos (DZ).

TIPO	PAR	FREQUÊNCIAS ESPERADAS	NÚMEROS OBSERVADOS
MZ	dom-dom	$pK^2 [2(1+p) - pK(4-K)]$	n_1
	dom-rec	$2pK(1-K) [2 - pK(3-K)]$	n_2
	rec-rec	$[1 - pK(2-K)]^2$	n_3
DZ	dom-dom	$\frac{K^2p(4+12p-8Kp+K^2p-8Kp^2+2K^2p^2+K^2p^3)}{4}$	n_4
	dom-rec	$\frac{Kp(8-4K-16Kp+8K^2p-K^3p+8K^2p^2-2K^3p^2-K^3p^3)}{2}$	n_5
	rec-rec	$(2-4Kp+K^2p+K^2p^2)^2/4$	n_6

As funções de verossimilhança correspondentes ao modelo I são, respectivamente, para gêmeos monozigóticos e dizigóticos:

$$L_{MZ} = (n_1+n_2) \cdot \log(p) + (2n_1+n_2) \cdot \log(K) + n_2 \cdot \log(1-K) \\ + n_1 \cdot \log[2(1+p) - pK(4-K)] + n_2 \cdot \log[2 - pK(3-K)] \\ + 2n_3 \cdot \log[1 - pK(2-K)]$$

$$L_{DZ} = n_4 \cdot \log[\frac{K^2p(4+12p-8Kp+K^2p-8Kp^2+2K^2p^2+K^2p^3)}{4}] \\ + n_5 \cdot \log[\frac{Kp(8-4K-16Kp+8K^2p-K^3p+8K^2p^2-2K^3p^2-K^3p^3)}{2}] \\ + n_6 \cdot \log[(2-4Kp+K^2p+K^2p^2)^2/4]$$

A frequência gênica (**p** ou **q**) e o valor da taxa de penetrância (**K**) que melhor explicam o conjunto de dados são estimados a partir das funções L_{MZ} e L_{DZ} .

O programa **TONGTWI1.BAS (ANEXO I)** estima os valores de **p** e **K** utilizando a amostra de pares gemelares monozigóticos

empregada por Praxedes & Otto (2000) em relação à característica capacidade de enrolar a língua, fornecendo os seguintes resultados:

N(DOM - DOM MZ TWIN PAIRS) = ? 138
N(DOM - REC MZ TWIN PAIRS) = ? 38
N(REC - REC MZ TWIN PAIRS) = ? 38
INITIAL GUESSES (P,K) = ? 0.5,0.8

MODEL I : $PEN(RR) = 1-(1-K)^2$, $PEN(Rr) = K$

p	s.e. (p)	K	s.e. (K)	dL/dp	dL/dK
0.50000	0.03275	0.80000	0.03004	107.04779	53.24016
0.59594	0.03462	0.81008	0.03064	2.88329	-4.51405
0.60152	0.03505	0.80448	0.03160	0.03857	-0.17641
0.60165	0.03506	0.80429	0.03163	0.00004	-0.00019
0.60165	0.03506	0.80429	0.03163	-0.00000	-0.00000
0.60165	0.03506	0.80429	0.03163	-0.00000	-0.00000

EXPLICIT SOLUTIONS : p = 0.60165
K = 0.80429

Como existem três classes de dados esperados (números de pares gemelares monozigóticos dominante-dominante, dominante-recessivo e recessivo-recessivo) e dois parâmetros (p e K) que são extraídos das amostras, as estimativas não podem ser testadas diretamente nas amostras das quais foram retiradas. Nesse caso, estimativas obtidas a partir de dados familiares para o modelo correspondente podem, por exemplo, ser utilizadas para viabilizar os testes.

No exemplo acima, os intervalos de confiança das estimativas de **p** e **K** obtidas de dados familiares e gemelares coincidem. De fato, o intervalo aproximado de confiança a 95% para as estimativas **p = 0.602** e **K = 0.804** são, respectivamente, **0.532-0.672** e **0.741-0.868**, enquanto que os obtidos através de dados familiares são **0.513-0.633** e **0.728-0.868**.

O método acima é aplicável apenas a características frequentes na população. Vários autores sugeriram métodos alternativos para estimar o parâmetro K a partir de um conjunto de dados de gêmeos monozigóticos apresentando características patológicas raras (Lasker, 1947; Allen, 1952; Pfändler, 1970).

Usando símbolos diferentes, esses autores obtiveram o parâmetro K direta ou indiretamente de $K = 2p_1/(1 + p_1)$ ou $K = 2n_1/(2n_1 + n_2)$, onde p_1 é a taxa de concordância e n_1 e n_2 são, respectivamente, os números observados de pares gemelares monozigóticos afetado-afetado e afetado-normal. As proporções esperadas de pares gemelares afetado-afetado e afetado-normal são dadas, respectivamente, por:

$$\begin{aligned} P(\text{af-af}) &= n_1/(n_1+n_2) = 2pqK^2/[2pqK^2+4pqK(1-K)] = K/(2-K), \text{ e} \\ P(\text{af-nl}) &= n_2/(n_1+n_2) = 4pqK(1-K)/[2pqK^2+4pqK(1-K)] \\ &= 2(1-K)/(2-K), \end{aligned}$$

de forma que a função de verossimilhança pode ser deduzida de

$$L_K = n_1 \cdot \log(K) + n_2 \cdot \log(1-K) - (n_1+n_2) \log(2-K),$$

com estimativa de máxima verossimilhança $K = 2n_1/(2n_1+n_2)$ e variância $\text{var}(K) = K(1-K)(2-K)^2/[2(n_1+n_2)]$. Executando o código Mathematica abaixo, obtemos o seguinte gráfico referente à função $K = f(p_1)$:

```
K = 2*p/(1+p);
Plot[K,{p,0,1}, Frame->True],
```

Os métodos de estimação da taxa de penetrância que empregam dados de gêmeos, no entanto, raramente se aplicam a

características raras na população, como é o caso de doenças ou defeitos genéticos. A probabilidade de compor uma amostra aleatória representativa de gêmeos concordantes e discordantes para uma dada condição genética é tão pequena, que inviabiliza a aplicação do método. Tomando como exemplo o retinoblastoma bilateral, uma doença autossômica dominante com penetrância incompleta, a frequência populacional estimada é da ordem de 1/10.000; considerando-se que a frequência de partos gemelares na maioria das populações é cerca de 1/100, a frequência esperada de gêmeos apresentando o tumor seria da ordem de 10^{-6} (Horimoto & Otto, 2008).

1.3) ESTIMATIVA DA TAXA DE PENETRÂNCIA UTILIZANDO INFORMAÇÕES GENEALÓGICAS

Um dos primeiros métodos registrados na literatura foi proposto por Stern (1973). Utiliza apenas as informações provenientes de indivíduos afetados e de heterozigotos comprovadamente não penetrantes e suas relações verticais na genealogia, não estabelecendo o grau de precisão da estimativa. Ou seja, a taxa de penetrância é estabelecida em termos da porcentagem de indivíduos que expressam fenotipicamente a doença ou defeito dentre todos aqueles que possuem o gene que condiciona a característica.

O método desenvolvido por Rogatko et al. (1986) considera todas as informações contidas na genealogia, atribuindo a cada indivíduo uma determinada probabilidade associada ao parâmetro K e fornece uma medida de incerteza da estimativa da taxa de penetrância. Essas propriedades o tornam mais adequado, quando comparado ao método de Stern (1973), pois possibilita a obtenção de uma estimativa de K mais precisa.

As genealogias empregadas na análise passam por uma filtragem, a qual retém apenas os indivíduos informativos ou relevantes à estimativa da taxa de penetrância.

Cada indivíduo que compõe a genealogia recebe, então, uma determinada função de K (que é probabilística quando não se ignoram os fatores constantes):

- a) indivíduos afetados (penetrantes): $K/2$ ou, ignorando-se fatores constantes, K ;
- b) indivíduos normais, heterozigotos certos para o gene que condiciona a anomalia (não penetrantes): $(1-K)/2$ ou, ignorando-se fatores constantes, $(1-K)$;
- c) indivíduos normais, sem filhos, descendentes diretos de heterozigotos certos: $1/2 + (1-K)/2 = (2-K)/2$, ou ignorando-se fatores constantes, $(2-K)$;
- d) árvores compostas por descendentes normais de um indivíduo heterozigoto certo: a função de K é construída de acordo com a estrutura das árvores e o número de gerações que as compõem;
- e) ramos derivados de uniões consanguíneas: uma função de probabilidades é associada a cada forma possível de segregação do alelo; a combinação dessas funções parciais determinará a função de verossimilhança correspondente à árvore com união consanguínea.

A combinação de todas as funções de probabilidades parciais listadas acima permite construir-se a função de verossimilhança (P_K) da genealogia.

A estimativa de máxima verossimilhança da taxa de penetrância K é obtida, então, através da solução da equação $dP_K/dK = 0$ [ou, mais convenientemente, $dL/dK = 0$, onde $L = \log(P_K)$]. O grau de incerteza do parâmetro K é avaliado mediante a construção do intervalo exato de credibilidade (bayesiano), determinado pela área que corresponde a 95% da área total da função de verossimilhança P_K .

2) BASES GENÉTICAS DA PENETRÂNCIA INCOMPLETA

Embora o entendimento dos mecanismos que determinam a penetrância incompleta tenha avançado progressivamente, ainda não se conhecem os processos envolvidos nesse fenômeno para a maioria das anomalias genéticas.

Dentre os mecanismos que explicam as bases genéticas da penetrância incompleta em doenças ou defeitos de herança monogênica estão: pré-mutação, ação epistática de genes modificadores, influência do alelo em *trans*, *imprinting* genômico, supressores de tumor e fatores ambientais (Zlotogora, 2003).

A ocorrência de gêmeos monozigóticos com fenótipos discordantes quanto à doenças dominantes poderia invalidar algumas dessas hipóteses sobre a penetrância incompleta. Entretanto, tem-se demonstrado mais recentemente que gêmeos monozigóticos não são tão geneticamente idênticos como se pensava (Hall, 1996; Gringas & Chen, 2001; Singh et al., 2002; Fraga et al., 2005; Bruder et al., 2008).

Abaixo, detalhamos alguns dos mecanismos genéticos responsáveis pela penetrância incompleta.

2.1) PRÉ-MUTAÇÃO

O genoma humano contém repetições em *tandem* de trinucleotídeos localizados em regiões codificadoras e não-codificadoras, que ocorrem de maneira polimórfica na população normal em grupos de cerca de 4 a 50 repetições (Cummings & Zoghbi, 2000). Quando esse número excede um certo limiar observam-se sinais clínicos associados a certas doenças.

O segmento de DNA expandido é altamente instável e tende a ser mais frequentemente amplificado do que contraído quando transmitido.

Quando amplificadas, essas repetições de trinucleotídeos apresentam um comportamento padrão, no qual a expansão através de gerações subsequentes correlaciona-se positivamente com a penetrância e a expressividade e

negativamente com a idade de manifestação da doença (Wilmot & Warren, 1998).

Manifestações clínicas mais graves e de instalação mais precoce a cada geração dizem respeito ao fenômeno conhecido como antecipação genética. A origem parental ou *imprinting* genômico do alelo mutante pode influenciar a ocorrência da antecipação em determinadas doenças monogênicas. Na doença de Huntington, por exemplo, o grau de antecipação em transmissões paternas é de 6 a 7 anos, enquanto que em transmissões maternas a antecipação observada é de apenas 1 ano (Paterson et al., 1998).

Os indivíduos assintomáticos que segregam segmentos expandidos de trinucleotídeos podem ser portadores de uma pré-mutação, ou seja, apresentarem um primeiro estágio da amplificação em que não existe expressão fenotípica da doença. Essa alteração genética, no entanto, pode tornar-se uma mutação completa ao ser transmitida pela linhagem germinativa.

O risco de recorrência é obtido, portanto, para indivíduos assintomáticos portadores da pré-mutação ou indivíduos levemente afetados portadores de segmentos de DNA pouco amplificados. Na distrofia miotônica (OMIM 160900), por exemplo, considerando-se o grau de parentesco com o propósito, o risco de doença na prole estará associado à probabilidade de indivíduos geralmente pouco afetados, com um número de cópias de trinucleotídeos próximo à faixa de pré-mutação, produzirem gametas com o segmento menos amplificado que o seu. Isso resultará em prole normal, porém portadora de uma amplificação próxima ao limiar e com risco alto de filhos portadores do segmento amplificado.

As doenças trinucleotídicas são classificadas em: (a) tipo I (doenças poliglutamínicas), caracterizadas pela amplificação do trinucleotídeo **CAG**; (b) tipo II, caracterizadas pela expansão dos trinucleotídeos **CTG**, **GAA**, **GCC** ou **CGG**. Todas elas comprometem o sistema nervoso.

As Tabelas 7 e 8, adaptadas de Cummings & Zoghbi (2000), apresentam as principais características de doenças trinucleotídicas desses dois tipos.

Tabela 7 - Doenças trinucleotídicas (poliglutamínicas) do tipo I.

DOENÇA	GENE	LOCO	Nº DE REPETIÇÕES	
			NORMAL	DOENÇA
Atrofia muscular espinobulbar	AR	Xq13-21	9-36	38-62
Doença de Huntington	HD	4p16.3	6-35	36-121
Síndrome de Haw-River	DRPLA	12p13.31	6-35	49-88
Ataxia espinocerebelar tipo 1	SCA1	6p23	6-44	39-82
Ataxia espinocerebelar tipo 2	SCA2	12q24.1	15-31	36-63
Ataxia espinocerebelar tipo 3 (Doença de Machado-Joseph)	SCA3 (MJD1)	14q32.1	12-40	55-84
Ataxia espinocerebelar tipo 6	SCA6	19p13	4-18	21-33
Ataxia espinocerebelar tipo 7	SCA7	13p12-13	4-35	37-306

Tabela 8 - Doenças trinucleotídicas do tipo II.

DOENÇA	GENE	LOCO	TRINUCLEOTÍDEO	NO. DE REPETIÇÕES	
				NORMAL	DOENÇA
Síndrome do cromossomo X frágil (FRAXA)	FRM1	Xq27.3	CGG	6-53	> 230
Síndrome do cromossomo X frágil (FRAXE)	FRM2	Xq28	GCC	6-35	> 200
Ataxia de Friedreich	X25	9q13-21.1	GAA	7-34	> 100
Distrofia miotônica	DMPK	19q13	CTG	5-37	> 50
Ataxia espinocerebelar tipo 8	SCA8	13q21	CTG	16-37	110 a 250
Ataxia espinocerebelar tipo 12	SCA12	5q31-33	CAG	7-28	66-78

2.2) AÇÃO EPISTÁTICA DE GENES MODIFICADORES

Genes modificadores constituem uma fonte importante de variação fenotípica e explicam satisfatoriamente a falta de correlação fenótipo-genótipo observada em determinadas anomalias genéticas, uma vez que interferem diretamente sobre a penetrância e expressividade do gene sobre o qual atuam.

Muitas doenças classificadas inicialmente como monogênicas são, na verdade, causadas ou moduladas pela ação de um pequeno número de locos, ou seja, essas doenças permanecem genéticas na sua etiologia, mas requerem a ação sinérgica de alelos mutantes localizados em locos distintos. Tais doenças são denominadas oligogênicas (Badano & Katsanis, 2002).

A Tabela 9 reúne alguns exemplos de doenças oligogênicas humanas, nas quais a penetrância incompleta dos genótipos é explicada pela ação dos genes modificadores.

Tabela 9 - Exemplos de doenças oligogênicas humanas.

Doença	Loco primário	Loco secundário
Doença de Hirschsprung (variante S-HSCR)	RET (10q11.2)	3p21 ^{d,e,f} ; 19q12 ^{d,e,f} ; 9q31 ^b
Doença de von Willebrand tipo I	VWF (12p13.3)	Grupo sanguíneo ABO ^{c,d}
Síndrome de van der Woude	VWS (1q32-41)	17p11.2 ^{a,d}

^aSertié et al.(1999); ^bBolk et al. (2000); ^cNitu-Whalley et al. (2000);

^dBadano & Katsanis (2002); ^eBolk et al. (2002); ^fHeyningen & Yeyati (2004).

A doença de Hirschsprung (HSCR - OMIM 142623), por exemplo, apresenta duas variantes de características genéticas distintas, identificadas de acordo com a extensão do segmento aganglionar do cólon: L-HSCR (segmento longo, forma sindrômica, observada em 20% dos indivíduos afetados) e S-HSCR (segmento curto, forma não-sindrômica, observada em 80% dos afetados). A penetrância do gene principal varia de acordo com a extensão do segmento aganglionar, com o sexo do afetado e com a ação dos genes modificadores, indicando que as mutações no loco primário

não são suficientes, por si só, para desencadear o quadro aganglionar (Bolk et al., 2000; Passarge, 2002).

2.3) INFLUÊNCIA DO ALELO EM TRANS

Um exemplo de doença que apresenta penetrância incompleta por esse mecanismo é a retinite pigmentosa (RP - OMIM 268000), uma anomalia geneticamente heterogênea. Em famílias com afetados por RP dominante com penetrância reduzida, indivíduos normais apresentam um polimorfismo bialélico do tipo selvagem que está significativamente correlacionado com o desenvolvimento da RP na descendência portadora da mutação. Quando em posição trans com a mutação do gene *RP11*, uma das formas do alelo selvagem é capaz silenciar a mutação, inibindo a manifestação da RP, enquanto a outra não (McGee et al., 1997; Rivolta et al., 2002).

2.4) IMPRINTING GENÔMICO

Nas síndromes de Angelman (OMIM 105830), Prader-Willi (OMIM 176270) e Beckwith-Wiedemann (OMIM 130650), por exemplo, a manifestação fenotípica é dependente da origem parental do gene mutante por *imprinting* genômico. A síndrome de Angelman é causada por mutações no gene *UBE3A* do cromossomo materno ou dissomia uniparental paterna do cromossomo 15. Somente indivíduos que recebem a mutação da mãe manifestam a síndrome; aqueles que recebem a mutação do pai não são afetados, mas, quando mulheres, podem transmitir a doença a sua progênie (Pasternak, 2007). A penetrância do genótipo está, portanto, relacionada à origem parental do gene mutante.

2.5) GENES SUPRESSORES DE TUMOR

Mutações na linhagem germinativa em locos supressores de tumor predis põem ao câncer familiar, embora seja necessária uma segunda mutação somática do alelo normal para o desenvolvimento do tumor. A ocorrência da segunda mutação é o fator que determina a penetrância nessas doenças.

O retinoblastoma (OMIM 180200), por exemplo, é um tumor ocular maligno que se desenvolve geralmente na primeira infância, seguindo um padrão de transmissão autossômico dominante com penetrância incompleta.

Situado no loco 13q14, o alelo normal do gene do retinoblastoma (*RBI*) é um antioncogene que, sob condições normais, tem a capacidade de interromper o ciclo de divisão celular pela ação de proteínas regulatórias da replicação do DNA, suprimindo a formação tumoral.

Nos casos herdados, a criança recebe de um dos pais o alelo mutante e tem a segunda cópia do alelo patológico gerada por um evento de mutação somática do alelo normal, o qual pode ocorrer em uma ou mais das cerca de um milhão de células da linhagem retinoblástica. Para casos não hereditários de retinoblastoma, um modelo proposto por Knudson (1971) explica a formação do retinoblastoma como resultado de dois eventos mutacionais somáticos em uma única célula (Knudson, 1986; Harbour, 2001)

O risco de recorrência dos casos hereditários de retinoblastoma (bilateral ou unilateral com recorrência familiar), estará associado, portanto, à taxa de penetrância do gene *RBI*, a qual corresponde à probabilidade de ocorrência do segundo evento mutacional no alelo normal.

3) SÍNDROME DA ECTRODACTILIA ASSOCIADA À HEMIMELIA TIBIAL (EHT)

Apresentamos uma revisão detalhada sobre essa síndrome, uma vez que genealogias com afetados por essa condição serão utilizadas de maneira abrangente como exemplos de aplicação de nossos modelos de penetrância.

A ectrodactilia (OMIM 183600) é uma malformação envolvendo os raios centrais das extremidades, com manifestações de sindactilia, fendas medianas e agenesia associada ou não à hipoplasia de falanges, metacarpos e metatarsos.

A hemimelia tibial (OMIM 275220) é uma anomalia geneticamente heterogênea caracterizada por deficiência no desenvolvimento embrionário da tíbia, geralmente sem comprometimento da fíbula. A manifestação do defeito pode ser uni ou bilateral, isolado ou associado a outras malformações esqueléticas e extra-esqueléticas, dando origem a diversas síndromes, dentre as quais a ectrodactilia associada à hemimelia tibial (OMIM 119100) (Majewski et al., 1985; Richieri-Costa et al., 1987; Naveed et al., 2006; Dayer et al., 2007).

A síndrome da ectrodactilia associada à hemimelia tibial (EHT) é uma condição rara que apresenta uma incidência estimada em aproximadamente 1:1.000.000 de nascidos vivos. Sua expressividade inter e intra-familiar é extremamente variável, determinando nos indivíduos afetados manifestações clínicas que variam de leves, como hálux hipoplásico a graves, como tetramonodactilia ou hemimelia transversa (Majewski et al., 1985).

Análises de ligação indicaram duas regiões genômicas relacionadas à expressão da EHT, localizadas, respectivamente, nos cromossomos 1q42 e 6q41. Regiões genômicas candidatas localizadas nos cromossomos 7q21 e 10q24, relacionadas originalmente à ectrodactilia, e 8q24.1, ligada à expressão da hemimelia tibial, foram excluídas em análises complementares (Marinoni et al., 1994; Naveed et al., 2007).

Nos casos familiares, a maioria dos trabalhos reporta herança autossômica dominante com penetrância incompleta (Majewski et al., 1985; Richieri-Costa et al., 1987; Spranger et

al., 1988; Sener et al., 1989; Marinoni et al., 1994; Shenoy et al., 2004; Naveed et al., 2006; Dayer et al., 2007). Entretanto, são também relatadas famílias com afetados provenientes de uniões consanguíneas ou concentrados em uma única irmandade, situações em que o modo de herança autossômico recessivo explicaria de forma mais adequada a manifestação da EHT (Mufti et al., 1987; Kohn, 1989; Majewski et al., 1996; Witters et al., 2001; Matsuyama et al., 2003), além de alguns poucos casos de herança possivelmente ligada ao cromossomo X (Ahmad et al., 1987; Faiyaz-Ul-Haque et al., 1993; Faiyaz-Ul-Haque et al., 2005).

Além disso, dois outros modelos foram propostos para explicar a ocorrência da síndrome: herança digênica e expansão da repetição de trinucleotídeos. No modelo de herança digênica proposto por Naveed et al. (2007), a ocorrência de indivíduos normais heterozigotos certos quanto ao alelo da EHT poderia ser explicada pela penetrância incompleta manifestando-se em ambos os locos ou, similar ao modelo proposto por Zlotogora & Nubani (1989) para ectrodactilia, um dos locos apresentaria a mutação dominante e o outro seria uma variante genômica responsável pelo controle da expressão. Já Zlotogora (1994) sugere que, a exemplo do que é observado nas síndromes do cromossomo X frágil, distrofia miotônica e coreia de Huntington, a base molecular da penetrância incompleta na EHT estaria relacionada à expansão de uma repetição de trinucleotídeos. Indivíduos normais heterozigotos certos quanto ao gene da EHT seriam portadores da pre-mutação instável que, ao expandir-se em gerações futuras, determinaria a manifestação clínica da síndrome. A gravidade dos defeitos estaria, então, relacionada ao número de cópias da trinca de nucleotídeos do alelo mutante.

O valor da taxa de penetrância estimado por Zlotogora (1994) em famílias com indivíduos afetados por ectrodactilia associada a outros defeitos de membro foi de 0.66. Este valor é muito superior à taxa de penetrância de 0.34 obtida por Maestrelli & Otto (1997), a partir da análise de uma família brasileira apresentando 12 afetados pela síndrome e dez heterozigotos certos quanto ao gene em questão. A discrepância

entre os valores pode ser explicada por diferenças metodológicas na estimativa do valor da taxa de penetração.

II) OBJETIVOS

O presente trabalho tem por objetivo principal o desenvolvimento de um programa computacional, em linguagem Microsoft® Visual Basic® 6.0, com interface gráfica que permite estimar o valor da taxa de penetrância a partir da análise da estrutura genealógica de famílias com casos de uma doença qualquer com herança autossômica dominante.

Os algoritmos empregados pelo programa basearam-se exclusivamente em idéias originais desenvolvidas por Rogatko et al. (1986) e por outros pesquisadores e estudantes do Laboratório de Genética Humana, geralmente sob orientação do Dr. Paulo A. Otto. Com a progressão do trabalho, defrontamo-nos com algumas situações inéditas que tivemos que resolver. Foram elas:

a) o desenvolvimento de métodos que permitam lidar com situações encontradas com certa frequência nos heredogramas publicados na literatura, como: (i) ausência de informações sobre o fenótipo do indivíduo heterozigoto certo gerador da genealogia; (ii) agrupamento de árvores de indivíduos normais sem a descrição detalhada da distribuição dos filhos entre os progenitores, por exemplo: registro de uma árvore de indivíduos normais com 7 progenitores e 39 filhos:

b) a proposta de um método alternativo ao descrito por Rogatko et al. (1986) para análise de estruturas da genealogia contendo uniões consanguíneas;

c) o desenvolvimento, por sugestão do Prof. Peter L. Pearson, de um modelo com taxa de penetrância variável, dependente do número de gerações do heredograma, uma vez que a inspeção de famílias com algumas doenças autossômicas dominantes (como é o caso da ectrodactilia associada à hemimelia tibial) sugere a existência

de um fenômeno similar à antecipação, em relação à taxa de penetrância;

d) a aplicação dos modelos com taxa de penetrância constante e variável a 21 heredogramas de famílias com indivíduos afetados pela síndrome da ectrodactilia associada à hemimelia tibial, com posterior realização de meta-análise;

e) a determinação de soluções gerais em forma analítica simples para (i) as funções de verossimilhança de qualquer árvore de indivíduos normais com ramificação regular e para (ii) as probabilidades de heterozigose de qualquer indivíduo pertencente a essas árvores;

f) o desenvolvimento, com a colaboração do graduando Márcio Teruo Onodera (autor principal), de uma versão do programa **PenCalc** para Internet, já em operação na rede. Em razão das dificuldades em desenvolver uma interface gráfica, usando a linguagem Microsoft® Visual Basic® 6.0, para exibir os resultados das probabilidades de heterozigose e o risco de afecção na prole para todos os indivíduos da genealogia, optamos por desenvolver esses dois módulos apenas nessa versão do programa para Internet, cuja linguagem de programação oferece uma interface mais intuitiva ao usuário.

Pretendíamos desenvolver o programa para lidar com situações de penetrância incompleta independentemente do modo de herança, ou seja, incluindo também os casos de herança ligada ao X e mitocondrial. O não-cumprimento desses objetivos deveu-se claramente à solução dos itens (a) a (f) indicados acima. Esses itens estão em desenvolvimento para fazer parte da versão final do programa.

III) MÉTODOS

1) MODELO

Adotamos no trabalho um modelo que pode ser representado pela seguinte matriz de probabilidades transicionais p_{ij} :

$$\begin{array}{cc} & \begin{array}{cc} dom & rec \end{array} \\ \begin{array}{c} a_1a_2 \\ a_2a_2 \end{array} & \begin{bmatrix} K & 1-K \\ 0 & 1 \end{bmatrix} \end{array}$$

Em doenças autossômicas dominantes, geralmente não se conhece o fenótipo dos indivíduos a_1a_1 . Assim, segundo o modelo, uma fração K dos indivíduos heterozigotos manifesta fenotipicamente a condição (penetrantes), enquanto uma fração $1-K$ dos heterozigotos a_1a_2 e todos os homozigotos a_2a_2 são normais, constituindo uma classe de indivíduos fenotipicamente indistinguíveis.

2) ESTIMATIVA DA TAXA DE PENETRÂNCIA

A estimativa da taxa de penetrância é obtida através da identificação de diferentes estruturas da genealogia filtrada e da composição de uma função de verossimilhança correspondente ao conjunto dessas estruturas.

2.1) ESTRUTURAS DA GENEALOGIA

As diferentes estruturas das genealogias pertinentes à estimativa da taxa de penetrância K , cada uma delas associada a uma função probabilística do parâmetro, são: indivíduo gerador da genealogia; indivíduo afetado; indivíduo normal, heterozigoto certo; indivíduo normal, sem filhos, descendente de um

heterozigoto certo; árvores de indivíduos normais; indivíduo resultante de união consanguínea; árvores com uniões consanguíneas; e gêmeos.

2.1.1) Indivíduo gerador da genealogia

É o indivíduo heterozigoto certo que inicia a genealogia filtrada. Se for afetado, a probabilidade a ele associada será **K**; se for normal, **(1-K)**. Em alguns casos, o fenótipo do indivíduo gerador da genealogia não é representado no heredograma, situação em que não lhe é atribuída nenhuma função probabilística de **K**.

Considerando que o gerador da genealogia é com certeza um heterozigoto certo, a chance a priori de um dos seus progenitores, que necessariamente são normais, ser heterozigoto é **1-K** (Frota-Pessoa et al., 1976; Otto & Maestrelli, 2000). Em ambos os lados da família (materno e paterno) do gerador existem vários outros indivíduos, ascendentes e colaterais, também normais. A ocorrência de todos esses familiares normais, excluindo-se a parte da genealogia descendente do gerador, faz com que a probabilidade final de heterozigose para os progenitores do gerador seja de uma ordem de grandeza desprezível, tornando provável que o primeiro afetado da família (o gerador da genealogia em estudo) seja produto de uma mutação nova ocorrida num dos gametas que o originaram.

2.1.2) Indivíduo afetado

A função de probabilidade associada a um indivíduo afetado é dada por

$$P(a_1a_2, \text{afetado} | a_1a_2 \times a_2a_2) = K/2$$

2.1.3) Indivíduo normal, heterozigoto certo

Nesse caso, a probabilidade é

$$P(a_1a_2, \text{normal} | a_1a_2 \times a_2a_2) = (1-K)/2$$

2.1.4) Indivíduo normal, sem filhos, descendente de um heterozigoto certo

Esse indivíduo é normal sob duas hipóteses: ou não recebeu o gene ou recebeu o gene e este não penetrou. A combinação dessas duas probabilidades indica a probabilidade final associada a essa estrutura:

$$P(\text{normal} | a_1a_2 \times a_2a_2) = 1/2 + (1-K)/2 = (2-K)/2$$

2.1.5) Árvores de indivíduos normais

São ramificações da genealogia compostas exclusivamente por indivíduos normais, descendentes de um heterozigoto certo. Não existe uma fórmula em forma analítica simples para descrever essas estruturas. A sua composição, entretanto, não traz maiores dificuldades, como se pode verificar através do seguinte exemplo de árvore de indivíduos normais

cujas funções de probabilidades são

$$P = 1/2 + (1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]\} \\ \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\}^2 \cdot [(2-K)/2] \quad .$$

2.1.6) Indivíduo resultante de união consanguínea

Um casamento consanguíneo qualquer pode favorecer a união de dois indivíduos heterozigotos. Nesse caso, dado que o homozigoto a_1a_1 é geralmente inviável ou letal, as probabilidades para a progênie homozigota normal e heterozigota penetrante e não-penetrante são, respectivamente:

$$P(a_2a_2|a_1a_2 \times a_1a_2) = 1/3$$

$$P(a_1a_2, \text{afetado}|a_1a_2 \times a_1a_2) = 2K/3$$

$$P(a_1a_2, \text{normal}|a_1a_2 \times a_1a_2) = 2(1-K)/3$$

2.1.7) Árvores com uniões consanguíneas

A função de probabilidades associada a esse tipo de árvore é composta pela combinação de todas as possíveis formas de segregação do gene (Rogatto et al., 1986). Exemplificamos abaixo a construção da função de probabilidades da seguinte estrutura consanguínea de uma genealogia da literatura (Marinoni et al., 1994):

As possíveis configurações genotípicas atribuíveis a essa árvore são:

As probabilidades associadas a essas configurações são dada por:

$$\begin{aligned}
P_1 &= 1/2 \cdot K/2 \cdot [(1-K)/2]^6 \cdot (2-K)/2 \\
P_2 &= 1/2 \cdot K/2 \cdot [(1-K)/2]^4 \cdot (2-K)/2 \\
P_3 &= 1/2 \cdot K/2 \cdot [(1-K)/2]^5 \cdot (2-K)/2 \\
P_4 &= 1/2 \cdot K/2 \cdot [(1-K)/2]^6 \cdot (2-K)/2 \\
P_5 &= K/2 \cdot [(1-K)/2]^6 \cdot 2(1-K)/3 \cdot (2-K)/2 \quad ,
\end{aligned}$$

sendo a função de probabilidades final correspondente à estrutura consanguínea:

$$\begin{aligned}
P &= P_1 + P_2 + P_3 + P_4 + P_5 \\
&= K/2 \cdot [(1-K)/2]^4 \cdot (2-K)/2 \cdot \{1/2 \cdot [(1-K)/2]^2 + 1/2 \\
&\quad + 1/2 \cdot (1-K)/2 + 1/2 \cdot [(1-K)/2]^2 + [(1-K)/2]^2 \cdot 2(1-K)/3\} \quad .
\end{aligned}$$

Com o objetivo de simplificar esses cálculos, empregamos um método alternativo no código do programa **PenCalc v. 1.0** (desenvolvido nesse trabalho), pelo qual a união consanguínea é desfeita, de forma que apenas um dos indivíduos (III.2 ou V.3), mas não ambos, transmita o gene a sua progênie. Aplicando o método à árvore acima, identificamos duas configurações possíveis:

configuração 1

configuração 2

com as correspondentes funções de probabilidades:

$$P_{\text{conf1}} = K_1 \cdot (1-K_1)^4 \cdot (2-K_1) \cdot \{1/2 + (1-K_1)/2 \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_1)/2]\}\}$$

$$P_{\text{conf2}} = K_2 \cdot (1-K_2)^6 \cdot (2-K_2)^2$$

Cada uma das funções acima é combinada à função de probabilidades obtida para o restante da genealogia gerando, assim, duas funções de verossimilhança distintas: uma que inclui a configuração 1 e outra, a configuração 2. Dois valores diferentes de K (K_1 e K_2) e de suas variâncias são obtidos. O valor da taxa de penetrância K é, então, obtido ponderando-se K_1 e K_2 pelo recíproco de suas variâncias:

$$K = (K_1/\text{var}K_1 + K_2/\text{var}K_2) / (1/\text{var}K_1 + 1/\text{var}K_2)$$

Em algumas genealogias que apresentam duas árvores consanguíneas, podemos identificar quatro configurações diferentes, que resultam das combinações possíveis de cada uma das duas configurações obtidas desfazendo-se a união consanguínea da primeira árvore com as duas correspondentes obtidas desfazendo-se a união da segunda árvore. Nesse caso, o valor final da estimativa de K é obtido da fórmula

$$K = (K_1/\text{var}K_1 + K_2/\text{var}K_2 + K_3/\text{var}K_3 + K_4/\text{var}K_4) / (1/\text{var}K_1 + 1/\text{var}K_2 + 1/\text{var}K_3 + 1/\text{var}K_4)$$

A Tabela 10 lista as estimativas de K obtidas empregando-se (1) o procedimento tradicional proposto por Rogatko et al. (1986) e (2) o método alternativo utilizado no programa **PenCalc v. 1.0**, para quatro diferentes genealogias da literatura contendo árvores com uniões consanguíneas. Ambas estimativas são similares, não se observando diferenças quando consideramos os resultados com dois algarismos significativos.

Tabela 10 - Estimativas da taxa de penetrância (K) obtidas empregando-se (1) o método proposto por Rogatko et al. (1986) e (2) o método alternativo utilizado no programa PenCalc v. 1.0, para quatro genealogias da literatura contendo árvores com uniões consanguíneas.

GENEALOGIA	ESTIMATIVAS DE K	
	(1)	(2)
Majewski et al., 1985	0,477	0,477
Sener et al., 1989	0,214	0,212
Marinoni et al., 1994	0,311	0,307
Naveed et al., 2006	0,344	0,342

2.1.8) Gêmeos

Pares gemelares de mesmo sexo descendentes de um portador certo do gene podem apresentar três composições fenotípicas distintas: a) ambos indivíduos normais; b) um indivíduo normal e um afetado; c) ambos afetados. Como anteriormente, a cada par é atribuída uma função de probabilidades, obtida pela aplicação formal do método bayesiano ou a partir da descrição de todas as possíveis composições genotípicas (método alternativo), como mostrado a seguir para as três situações (a, b, c) descritas acima, levando-se em conta que os gêmeos de mesmo sexo podem ser monozigóticos (MZ) ou dizigóticos (DZ).

a) ambos os gêmeos normais:

Método bayesiano

Probabilidades	MZ	DZ
A priori	1/3	2/3
Condicional	$1/2 + 1/2 \cdot (1-K)^2$	$[(2-K)/2]^2$
Conjunta	$1/3 \cdot [1/2 + 1/2 \cdot (1-K)^2]$	$2/3 \cdot [(2-K)/2]^2$

$$P(\text{ambos normais} | \text{MZ ou DZ}) = \frac{1/3 \cdot [1/2 + (1-K)^2/2] + 2/3 \cdot [(2-K)/2]^2}{1/3 \cdot [1/2 + (1-K)^2/2] + 2/3 \cdot [(2-K)/2]^2} = \frac{(3-3K+K^2)}{3}$$

Método Alternativo

MZ (1/3)

DZ (2/3)

$$P(\text{ambos normais} | \text{MZ ou DZ}) = \frac{1/3 \cdot [1/2 + (1-K)^2/2] + 2/3 \cdot [(2-K)/2]^2}{1/3 \cdot [1/2 + (1-K)^2/2] + 2/3 \cdot [(2-K)/2]^2} = \frac{(3-3K+K^2)}{3}$$

b) um afetado e um normal:

Método bayesiano

Probabilidades	MZ	DZ
A priori	1/3	2/3
Condicional	$1/2 \cdot K \cdot (1-K)$	$1/2 \cdot K \cdot (1/2 + (1-K)/2)$
Conjunta	$K \cdot (1-K) / 6$	$K \cdot (2-K) / 6$

$$\begin{aligned}
 P(\text{um normal e um afetado} | \text{MZ ou DZ}) &= K \cdot (1-K) / 6 + K \cdot (2-K) / 6 \\
 &= (3K - 2K^2) / 6 = K \cdot (3 - 2K) / 6
 \end{aligned}$$

Método Alternativo

MZ (1/3)

 $K(1-K) / 2$

DZ (2/3)

 $K/4$

 $K(1-K) / 4$

$$\begin{aligned}
 P(\text{um normal e um afetado} | \text{MZ ou DZ}) &= K \cdot (1-K) / 6 + K \cdot (2-K) / 6 \\
 &= (3K - 2K^2) / 6 = K \cdot (3 - 2K) / 6
 \end{aligned}$$

c) ambos afetados:

Método bayesiano

Probabilidades	MZ	DZ
A priori	1/3	2/3
Condicional	$1/2 \cdot K^2$	$1/2 \cdot 1/2 \cdot K^2$
Conjunta	$K^2/6$	$K^2/6$

$$P(\text{ambos afetados} | \text{MZ ou DZ}) = K^2/6 + K^2/6 = K^2/3$$

Método Alternativo

MZ (1/3)

$1/2 \cdot K^2$

DZ (2/3)

$1/4 \cdot K^2$

$$P(\text{ambos afetados} | \text{MZ ou DZ}) = K^2/6 + K^2/6 = K^2/3$$

2.2) FILTRAGEM DA GENEALOGIA

A estrutura da genealogia é inicialmente filtrada, de forma a manter apenas os indivíduos informativos para a estimativa da taxa de penetrância, como no exemplo abaixo:

2.3) FUNÇÃO DE VEROSSIMILHANÇA

A função de verossimilhança correspondente à genealogia filtrada é obtida multiplicando-se todas as funções probabilísticas de **K** associadas a cada uma das estruturas.

Na genealogia filtrada

por exemplo, identificamos as seguintes estruturas:

Tabela 11 - Probabilidades e números de ocorrências das estruturas presentes na genealogia descrita acima.

ESTRUTURA		PROBABILIDADE	NÚMERO DE OCORRÊNCIAS
afetado	$K/2^*$	4
heterozigoto certo, normal	$(1-K)/2^*$	4
normal, sem filhos, descendente de heterozigoto certo	$(2-K)/2$	3
árvore normal de descendentes de heterozigoto certo	$1/2 + (1-K)/2 \cdot [(2-K)/2]^2$	1

* No caso do indivíduo heterozigoto certo, afetado e normal, ser um gerador de genealogia, as probabilidades são respectivamente K e 1-K.

A função de verossimilhança dessa genealogia, ignorando-se as quantidades constantes, é dada por

$$P_K = K^4 \cdot (1-K)^4 \cdot (2-K)^3 \cdot [4 + (1-K)(2-K)^2] \quad .$$

2.4) MÉTODO DE MÁXIMA VEROSSIMILHANÇA

A estimativa da taxa de penetrância K , obtida mediante a aplicação do *método de máxima verossimilhança*, corresponde ao valor do parâmetro K que maximiza a função de verossimilhança P_K . Intuitivamente, o valor estimado é o que melhor explica a ocorrência da genealogia observada.

Como a função P_K é um produto de funções de probabilidades do parâmetro K , a sua diferenciação (aplicada para obter-se a estimativa em $dP_K/dK = 0$) é complicada. Por essa razão, é conveniente trabalhar-se com o logaritmo da função ($L_K = \log P_K$), cujo extremo no intervalo (0,1) do argumento K vai coincidir com o extremo da função original.

No caso de funções com mais de um parâmetro ($K_1, K_2, \dots, K_i$), a forma logarítmica da função de verossimilhança é diferenciada em relação aos diversos argumentos, obtendo-se em seguida o sistema de equações $\{\partial L/\partial K_1 = 0, \partial L/\partial K_2 = 0, \partial L/\partial K_3 = 0, \partial L/\partial K_4 = 0\}$, cuja solução fornece as estimativas $K_1, \dots, K_4$.

O valor estimado do parâmetro K ou dos parâmetros $K_1, K_2, \dots, K_i$, na situação acima, são obtidos por meio da aplicação de métodos numéricos de localização de zeros da função num intervalo qualquer do argumento, como o *generalizado de Newton-Raphson*.

No caso da genealogia apresentada no item 2.3, empregando o *método de máxima verossimilhança*, obtemos a estimativa de $K = 0,418$, utilizando, por exemplo, o seguinte código Mathematica:

```
L = 4 * Log[K] + 4 * Log[1-K] + 3 * Log[2-K] + Log[4+(1-K)*(2-K)^2];
dLdK = D[L,K];
Plot[{L/100,dLdK/100},{K,0,1}, PlotRange->{1,-1},
      AspectRatio->1, Frame->True]
FindRoot[dLdK==0,{K,0.5}]
```

```
{K -> 0.418115}
```


Essa estimativa da taxa de penetrância (**K = 0,418**) corresponde ao valor de K que maximiza a função logarítmica, valor esse que coincide com o ponto em que a derivada corta o eixo das abscissas.

3) INTERVALO EXATO DE CONFIANÇA A 95%

O intervalo exato de confiança a 95% é uma medida de confiabilidade para a estimativa de K, definido como a área de $P_K = f(K)$ entre dois valores de K ($K = a$ e $K = b$) maiores que 0 e menores que 1 com valores idênticos de ordenada $[L(K=a) = L(K=b)]$, de forma que $\int_a^b [f(K)dK] / \int_0^1 [f(K)dK] = 0,95$. Para a função acima, esses limites são respectivamente **0.163** e **0.725**, obtidos através de programas como o **CONFINT3.BAS (ANEXO I)**, que emprega uma modificação do método *osculatório de Romberg*, o qual determina numericamente o valor da área de uma integral definida.

A representação gráfica desse intervalo exato de credibilidade pode ser obtida através de códigos, como o Mathematica abaixo:

```
a=0.16279;
b=0.72477;
c=0.41811;
fa= a^4*(1-a)^4*(2-a)^3*(4+(1 - a)*(2 - a)^2);
fb= b^4*(1-b)^4*(2-b)^3*(4+(1 - b)*(2 - b)^2);
fc= c^4*(1-c)^4*(2-c)^3*(4+(1 - c)*(2 - c)^2);
Show[
  Plot[K^4*(1-K)^4*(2-K)^3*(4+(1 - K)*(2 - K)^2), {K,0,1},
    Frame->True,
    DisplayFunction -> Identity],
  Graphics[{
    Line[{{a,0},{a,fa}}],
    Line[{{c,0},{c,fc}}],
    Line[{{b,0},{b,fb}}],
  }],
  DisplayFunction -> $DisplayFunction,
  AspectRatio -> 1/2];
```

4) PROBABILIDADE DE HETEROZIGOSE E CÁLCULO DE RISCO

O cálculo da probabilidade de heterozigose não oferece maiores dificuldades. Obviamente, o valor dessa probabilidade é 1 para indivíduos portadores certos do gene (afetados ou não). Não existe uma fórmula geral sob forma analítica simples para as probabilidades de heterozigose de indivíduos pertencentes às árvores de indivíduos normais, descendentes de um portador certo do gene, entretanto, podemos determiná-las aplicando princípios do cálculo de probabilidades, como mostrado a seguir.

4.1) ÁRVORES DE INDIVÍDUOS NORMAIS COM UMA GERAÇÃO

Árvores desse tipo equivalem a indivíduos normais sem filhos. A função de probabilidades associada a esse indivíduo é $P_{K1} = 1/2 + (1-K)/2 = (2-K)/2$ ou, considerando-se $B_1 = (1-K)/2$, $P_{K1} = 1/2 + B_1$. A probabilidade de heterozigose é, portanto, dada por

$$P(\text{het}) = B_1 / 1/2 + B_1 = (1-K) / (2-K)$$

4.2) ÁRVORES DE INDIVÍDUOS NORMAIS COM DUAS GERAÇÕES

A função de probabilidades referente a essa árvore é $P_{K2} = 1/2 + (1-K)/2 \cdot [(2-K)/2]^n = 1/2 + B_2$, sendo $B_2 = (1-K)/2 \cdot [(2-K)/2]^n$.

A probabilidade de heterozigose do indivíduo da primeira geração, calculada de maneira semelhante ao exemplo anterior, é $P(het) = B_2 / (1/2 + B_2)$.

Os n indivíduos da segunda geração apresentam probabilidades de heterozigose idênticas, as quais podem ser calculadas a partir das probabilidades a priori (probabilidades de heterozigose $[P_p(het)]$ e homozigose do progenitor tendo $n-1$ filhos) e condicionais de normalidade, como mostrado abaixo:

	Heterozigose	Homozigose
a priori	$P_p(het)/2$	$1 - P_p(het)/2$
condicional (normal)	$(1-K)$	1
conjunta	$P_p(het)/2 \cdot (1-K)$	$1 - P_p(het)/2$

Normalizando as quantidades finais acima, obtemos as probabilidades de heterozigose para os indivíduos dessa geração:

$$P(het) = P_p(het)/2 \cdot (1-K) / \{P_p(het)/2 \cdot (1-K) + [1 - P_p(het)/2]\}$$

Aplicando-se elementos do cálculo bayesiano, como detalhado acima, podemos obter as probabilidades de heterozigose para indivíduos pertencentes à árvores normais com um número qualquer de gerações.

Uma vez calculada a probabilidade de heterozigose, o risco de afecção da prole é calculado diretamente da fórmula

$$R = P(het) \cdot 1/2 \cdot K = P(het) \cdot K/2$$

5) TESTE DE RAZÃO DE VEROSSIMILHANÇAS (TESTE G)

Esse teste foi empregado para avaliar se o ajuste do modelo com taxa de penetrância variável aos dados observados era melhor que o obtido com o modelo com taxa de penetrância constante.

Sendo L_A a função de verossimilhança de um modelo qualquer sob uma hipótese qualquer A (H_A) e L_B a função do modelo sob a hipótese alternativa B (H_B), a razão de verossimilhanças (Λ) é dada por

$$\Lambda = \frac{\max[L_A(H_A|\text{dados})]}{\max[L_B(H_B|\text{dados})]},$$

e a estatística do teste definida como

$$\begin{aligned} G &= -2.\log(\Lambda) = -2.[\log(L_A) - \log(L_B)] \\ &= -2.\log L_A/L_B = 2.\log L_B/L_A \end{aligned}$$

A estatística $-2.\log(\Lambda)$ corresponde assintoticamente a uma distribuição χ^2 com graus de liberdade igual à diferença entre os números de parâmetros dos modelos A e B contrastados. Um valor do teste significativo indica que o modelo sob a hipótese B adere de maneira mais adequada aos dados observados que o modelo sob a hipótese A.

6) ALGUNS RESULTADOS GERAIS DE INTERESSE TEÓRICO

6.1) FÓRMULAS GERAIS PARA ÁRVORES DE INDIVÍDUOS NORMAIS COM RAMIFICAÇÃO REGULAR

As fórmulas para as árvores de indivíduos normais descendentes de portadores certos do gene não possuem uma representação geral em forma analítica simples. A determinação da probabilidade correspondente a qualquer árvore, apesar de trabalhosa, é facilmente conseguida aplicando-se elementos básicos da teoria de probabilidades. Por exemplo, dada a árvore

a função de probabilidades a ela associada é

$$\begin{aligned}
 P &= 1/2 + (1-K)/2 \\
 &\cdot \{1/2 + (1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\} \cdot [(2-K)/2]\} \\
 &\cdot [(2-K)/2] \cdot \{1/2 + (1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]\}\} \\
 &\cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\} \cdot
 \end{aligned}$$

Podemos, no entanto, obter facilmente fórmulas gerais em forma analítica simples para árvores de indivíduos normais com ramificação regular.

Nas fórmulas que se seguem, P_{ki} indica a função de verossimilhança de uma árvore contendo indivíduos normais em i gerações; K é o valor da taxa de penetrância; $P_{ij}(\text{het})$ indica a probabilidade de heterozigose de um indivíduo pertencente à j -ésima geração de uma árvore de indivíduos normais com i gerações (obviamente, $j \leq i$).

Detalharemos o método apenas para os casos de árvores com 0 e 1 ramificações por geração, casos esses que já permitem que se deduza o comportamento geral das expressões correspondentes a qualquer árvore com **n** ramificações por geração.

As árvores regulares possíveis com 0 ramificações por geração (árvores sem ramificações, constituídas por indivíduos com apenas um filho) são, considerando-se 1, 2, 3 e 4 gerações:

Fazendo-se **a = 1/2** e **b = (1-K)/2**, as funções de verossimilhança correspondentes a essas árvores tomam respectivamente as formas:

$$P_{K1} = a + b$$

$$P_{K2} = a + b(a + b)$$

$$P_{K3} = a + b[a + b(a + b)]$$

$$P_{K4} = a + b\{a + b[a + b(a + b)]\} \quad .$$

Dessas fórmulas, obtemos as probabilidades de heterozigose para os indivíduos da primeira geração, que são dadas, respectivamente, por:

$$P_{11}(\text{het}) = b / (a + b)$$

$$P_{21}(\text{het}) = b(a + b) / [a + b(a + b)]$$

$$P_{31}(\text{het}) = b[a + b(a + b)] / \{a + b[a + b(a + b)]\}$$

$$P_{41}(\text{het}) = b\{a + b[a + b(a + b)]\} / \{a + b\{a + b[a + b(a + b)]\}\}.$$

Aplicando-se elementos do método bayesiano, obtemos as probabilidades de heterozigose também para os indivíduos da segunda geração das árvores de 2, 3 e 4 gerações, as probabilidades de heterozigose para os indivíduos da terceira geração das árvores de 3 e 4 gerações e a probabilidade de heterozigose para os indivíduos da quarta geração da árvore de 4 gerações:

$$P_{22}(\text{het}) = b^2 / [a + b(a + b)]$$

$$P_{32}(\text{het}) = b^2(a + b) / \{a + b[a + b(a + b)]\}$$

$$P_{42}(\text{het}) = b^2[a + b(a + b)] / \{a + b\{a + b[a + b(a + b)]\}\}$$

$$P_{33}(\text{het}) = b^3 / \{a + b[a + b(a + b)]\}$$

$$P_{43}(\text{het}) = b^3(a + b) / \{a + b\{a + b[a + b(a + b)]\}\}$$

$$P_{44}(\text{het}) = b^4 / \{a + b\{a + b[a + b(a + b)]\}\} \quad .$$

Da inspeção dessas fórmulas verifica-se que

$$\begin{aligned} P_{22}(\text{het}) &= P_{11}(\text{het}) \cdot P_{21}(\text{het}) \\ &= b / (a + b) \cdot b(a + b) / [a + b(a + b)] \\ &= b^2 / [a + b(a + b)] \end{aligned}$$

e também que

$$\begin{aligned} P_{32}(\text{het}) &= P_{31}(\text{het}) \cdot P_{21}(\text{het}) \\ P_{33}(\text{het}) &= P_{31}(\text{het}) \cdot P_{21}(\text{het}) \cdot P_{11}(\text{het}) \\ P_{42}(\text{het}) &= P_{41}(\text{het}) \cdot P_{31}(\text{het}) \\ P_{43}(\text{het}) &= P_{41}(\text{het}) \cdot P_{31}(\text{het}) \cdot P_{21}(\text{het}) \\ P_{44}(\text{het}) &= P_{41}(\text{het}) \cdot P_{31}(\text{het}) \cdot P_{21}(\text{het}) \cdot P_{11}(\text{het}) \quad . \end{aligned}$$

Levando-se em conta todos os resultados acima, a expressão geral para $P_{ij}(\text{het})$, em que i é a profundidade da árvore e $j = 1, 2, 3, \dots$, i é a geração à qual pertence o indivíduo, é dada por:

$$P_{ij}(het) = \prod_{k=i-j+1}^i P_{k1}(het)$$

Isso tudo deixa claro que para se obter a probabilidade de heterozigose para qualquer indivíduo de uma árvore regular basta conhecer os valores da probabilidade de heterozigose dos indivíduos da primeira geração. Como vimos acima, essas probabilidades tomam os valores

$$P_{11}(het) = b / (a + b)$$

$$P_{21}(het) = b(a + b) / [a + b(a + b)]$$

$$P_{31}(het) = b[a + b(a + b)] / \{a + b[a + b(a + b)]\}$$

$$P_{41}(het) = b\{a + b[a + b(a + b)]\} / \{a + b\{a + b[a + b(a + b)]\}\},$$

tirados diretamente das funções de verossimilhança das árvores correspondentes:

$$P_{K1} = a + b$$

$$P_{K2} = a + b(a + b)$$

$$P_{K3} = a + b[a + b(a + b)]$$

$$P_{K4} = a + b\{a + b[a + b(a + b)]\} \quad .$$

As expressões acima são formas aninhadas dos elementos da série geométrica:

$$\begin{aligned} & a + b, a + ab + b^2, a + ab + ab^2 + b^3, \\ & a + ab + ab^2 + ab^3 + b^4, \dots, a(b^0 + b^1 + b^2 + \dots + b^{i-1}) + b^i \\ & = a \cdot \sum_{k=0}^{i-1} b^k + b^i = a(1-b^i)/(1-b) + b^i \quad , \end{aligned}$$

sendo a probabilidade de heterozigose P_{i1} para o indivíduo da primeira geração ($j = 1$) de uma árvore não ramificada com i gerações dada pela expressão

$$\begin{aligned} P_{i1}(het) &= [a(1-b^i)/(1-b) + b^i - a] / [a(1-b^i)/(1-b) + b^i] \\ &= [a(b-b^i)/(1-b) + b^i] / [a(1-b^i)/(1-b) + b^i] \\ &= b[a(1-b^{i-1})/(1-b) + b^{i-1}] / [a(1-b^i)/(1-b) + b^i] \quad . \end{aligned}$$

A partir dessa fórmula, obtemos as fórmulas de probabilidade de heterozigose para qualquer indivíduo da genealogia. Por exemplo, a probabilidade de heterozigose $P_{43}(\text{het})$ para o indivíduo da terceira geração de uma árvore não ramificada de quatro gerações é dada por

$$\begin{aligned}
 P_{43}(\text{het}) &= P_{41}(\text{het}) \cdot P_{31}(\text{het}) \cdot P_{21}(\text{het}) \\
 &= [a(b-b^4)/(1-b) + b^4] / [a(1-b^4)/(1-b) + b^4] \\
 &\quad \cdot [a(b-b^3)/(1-b) + b^3] / [a(1-b^3)/(1-b) + b^3] \\
 &\quad \cdot [a(b-b^2)/(1-b) + b^2] / [a(1-b^2)/(1-b) + b^2] \\
 &= b[a(1-b^3)/(1-b) + b^3] / [a(1-b^4)/(1-b) + b^4] \\
 &\quad \cdot b[a(1-b^2)/(1-b) + b^2] / [a(1-b^3)/(1-b) + b^3] \\
 &\quad \cdot b[a(1-b^1)/(1-b) + b^1] / [a(1-b^2)/(1-b) + b^2] \\
 &= b^3(a+b) / [a(1-b^4)/(1-b) + b^4] \\
 &= b^3(a+b) / [a(1+b^2+b^3) + b^4] \\
 &= b^3(a+b) / \{a + b[a + b[a + b(a+b)]]\} .
 \end{aligned}$$

Consideraremos em seguida o caso de árvores dicotômicas, ou seja, com uma ramificação por geração (cada indivíduo da árvore tem exatamente dois filhos). Como no caso anterior, mostramos abaixo as configurações correspondentes à árvores com 1, 2, 3 e 4 gerações.

As expressões de probabilidade correspondentes a essas quatro árvores são, respectivamente:

$$P_{K1} = a + b$$

$$P_{K2} = a + b(a + b)^2$$

$$P_{K3} = a + b[a + b(a + b)^2]^2$$

$$P_{K4} = a + b\{a + b[a + b(a + b)^2]^2\}^2 \quad .$$

Como no caso anterior das árvores não ramificadas, das fórmulas acima obtemos as probabilidades de heterozigose para os indivíduos da primeira geração, que são dadas respectivamente por:

$$P_{11}(\text{het}) = b / (a + b)$$

$$P_{21}(\text{het}) = b(a + b)^2 / [a + b(a + b)^2]$$

$$P_{31}(\text{het}) = b[a + b(a + b)^2]^2 / \{a + b[a + b(a + b)^2]^2\}$$

$$P_{41}(\text{het}) = b\{a + b[a + b(a + b)^2]^2\}^2 / \{a + b\{a + b[a + b(a + b)^2]^2\}^2\} \quad .$$

Os resultados gerais obtidos no caso anterior são imediatamente aplicáveis à situação das árvores dicotômicas. De fato, empregando-se elementos do cálculo de probabilidades, determinamos, por exemplo, que a probabilidade de heterozigose $P_{32}(\text{het})$ para um indivíduo qualquer da segunda geração de uma árvore dicotômica de 3 gerações é

$$P_{32}(\text{het}) = b^2(a + b)^2[a + b(a + b)^2] / \{a + b[a + b(a + b)^2]^2\}$$

e essa expressão é igual ao produto $P_{31}(\text{het}) \cdot P_{21}(\text{het})$, exatamente como no caso anterior das árvores não-ramificadas:

$$\begin{aligned} P_{31}(\text{het}) \cdot P_{21}(\text{het}) &= b[a + b(a + b)^2]^2 / \{a + b[a + b(a + b)^2]^2\} \\ &\quad \cdot b(a + b)^2 / [a + b(a + b)^2] \\ &= b^2(a + b)^2[a + b(a + b)^2] \\ &\quad / \{a + b[a + b(a + b)^2]^2\} \\ &= P_{32}(\text{het}) \quad . \end{aligned}$$

Também como no caso anterior, expressões gerais de forma não aninhada podem ser obtidas dessas fórmulas para árvores com um número qualquer de ramificações regulares por

geração (cada indivíduo da árvore tendo n filhos). Nesse caso, as fórmulas para $P_{K1}, \dots, P_{K4}$ reduzem-se a:

$$P_{K1} = a + b$$

$$P_{K2} = a + b(a + b)^n$$

$$P_{K3} = a + b[a + b(a + b)^n]^n$$

$$P_{K4} = a + b\{a + b[a + b(a + b)^n]^n\}^n \quad .$$

Podemos, portanto, deduzir uma única fórmula aplicável à probabilidade de heterozigose para um indivíduo qualquer pertencente a **j-ésima** geração de uma árvore regular com **n** ramificações por geração com **i** gerações de profundidade. Como essas árvores raramente ocorrerem nas genealogias, o assunto possui apenas interesse teórico.

6.2) ÁRVORES NORMAIS COM UNIÕES CONSANGUÍNEAS

Mostramos abaixo alguns resultados correspondendo à situação em que o indivíduo da última geração é a progênie de um casal aparentado biologicamente.

A dedução das fórmulas é mostrada com detalhes para os casos de filhos de uniões entre irmãos, tios-sobrinhas (ou tias-sobrinhos) e casais de primos em primeiro grau.

No caso de filho de irmãos, as configurações genotípicas possíveis das árvores de indivíduos normais são as seguintes, em que os símbolos assinalados por um ponto indicam indivíduos heterozigotos não-penetrantes e as setas indicam a existência de uma configuração extra especular, não incluída entre os desenhos.

A função de probabilidades associada a cada configuração é dada por:

$$\begin{aligned}
 P_{(1)} &= (1/2)^2 \\
 P_{(2)} &= 2(1-K)/2 \cdot (1/2)^2 \\
 P_{(3)} &= 2[(1-K)/2]^2 \cdot 1/2 \\
 P_{(4)} &= [(1-K)/2]^2 \cdot 1/3 \\
 P_{(5)} &= [(1-K)/2]^2 \cdot 2(1-K)/3 \quad ,
 \end{aligned}$$

sendo a função de probabilidades dessa árvore

$$\begin{aligned}
 P_{1K} = \sum P_{(i)} &= P_{(1)} + \dots + P_{(5)} \\
 &= (12-17K+10K^2-2K^3) / 12 \quad .
 \end{aligned}$$

Portanto, as probabilidades $P(\text{II.1} = \text{nl}, \text{aa})$ e $P(\text{II.1} = \text{nl}, \text{Aa})$ de homozigose e heterozigose para o indivíduo II-1 (progênie normal do casal de irmãos) são respectivamente

$$\begin{aligned}
 P(\text{II.1} = \text{nl}, \text{aa}) &= [P_{(1)} + P_{(2)} + P_{(4)}] / \sum P_{(i)} \\
 &= \{ [3+3(1-K)+(1-K)^2] / 12 \} / P_{1K} \\
 &= (7-5K+K^2) / (12-17K+10K^2-2K^3)
 \end{aligned}$$

e

$$\begin{aligned}
P(\text{II.1} = n1, Aa) &= [P_{(3)} + P_{(5)}] / \sum P_{(i)} \\
&= \{ [3(1-K)^2 + 2(1-K)^3] / 12 \} / P_{1K} \\
&= (5 - 12K + 9K^2 - 2K^3) / (12 - 17K + 10K^2 - 2K^3) .
\end{aligned}$$

Na situação de filho de casais tio-sobrinha (ou tia-sobrinho), as configurações genotípicas possíveis das árvores de indivíduos normais são:

sendo a função de probabilidades associada a cada configuração dada por:

$$\begin{aligned}
P_{(1)} &= (1/2)^2 \\
P_{(2)} &= (1-K)/2 \cdot (1/2)^2 \\
P_{(3)} &= (1-K)/2 \cdot (1/2)^2 \\
P_{(4)} &= [(1-K)/2]^2 \cdot 1/2 \\
P_{(5)} &= [(1-K)/2]^2 \cdot (1/2)^2 \\
P_{(6)} &= [(1-K)/2]^3 \cdot 1/2 \\
P_{(7)} &= [(1-K)/2]^2 \cdot (1/2)^2 \\
P_{(8)} &= [(1-K)/2]^3 \cdot 1/2 \\
P_{(9)} &= [(1-K)/2]^3 \cdot 1/3 \\
P_{(10)} &= [(1-K)/2]^3 \cdot 2(1-K)/3
\end{aligned}$$

A função de probabilidades dessa árvore é, portanto,

$$\begin{aligned}
P_{2K} &= \sum P_{(i)} = P_{(1)} + \dots + P_{(10)} \\
&= 1/4 + (1-K)/8 + (1-K)(2-K)/8 + (1-K)^2 \cdot (2-K)/8 \\
&\quad + (1-K)^3 \cdot (3-2K)/24 \\
&= (24-38K+30K^2-12K^3+2K^4) / 24
\end{aligned}$$

e as probabilidades de homozigose e heterozigose do indivíduo III.1 são, respectivamente:

$$\begin{aligned}
P(\text{III.1} = n1, aa) &= [P_{(1)} + P_{(2)} + P_{(3)} + P_{(5)} + P_{(7)} + P_{(9)}] / \sum P_{(i)} \\
&= \{ [6 + 6(1-K) + 3(1-K)^2 + (1-K)^3] / 24 \} / P_{2K} \\
&= (16-15K+6K^2-K^3) \\
&\quad / (24-38K+30K^2-12K^3+2K^4)
\end{aligned}$$

e

$$\begin{aligned}
P(\text{III.1} = n1, Aa) &= [P_{(4)} + P_{(6)} + P_{(8)} + P_{(10)}] / \sum P_{(i)} \\
&= [3(1-K)^2 + 3(1-K)^3 + 2(1-K)^4] / 24 \} / P_{2K} \\
&= (8-23K+24K^2-11K^3+2K^4) \\
&\quad / (24-38K+30K^2-12K^3+2K^4) .
\end{aligned}$$

No caso de filho de primos em primeiro grau, as configurações genotípicas possíveis das árvores de indivíduos normais são:

As probabilidades correspondentes às nove configurações acima são:

$$\begin{aligned}
 P_{(1)} &= (1/2)^2 \\
 P_{(2)} &= 2(1-K)/2 \cdot (1/2)^2 \\
 P_{(3)} &= [(1-K)/2]^2 \cdot (1/2)^2 \\
 P_{(4)} &= 2[(1-K)/2]^2 \cdot (1/2)^2 \\
 P_{(5)} &= 2[(1-K)/2]^3 \cdot (1/2)^2 \\
 P_{(6)} &= [(1-K)/2]^4 \cdot 1/3 \\
 P_{(7)} &= 2[(1-K)/2]^3 \cdot 1/2 \\
 P_{(8)} &= 2[(1-K)/2]^4 \cdot 1/2 \\
 P_{(9)} &= [(1-K)/2]^4 \cdot 2(1-K)/3
 \end{aligned}$$

A função de probabilidades associada a essa árvore é dada por

$$\begin{aligned}
 P_{3K} = \sum P_{(i)} &= P_{(1)} + \dots + P_{(9)} \\
 &= 1/4 + (1-K)(5-K)/16 + (1-K)^2 \cdot (2-K)(3-K)/16 \\
 &\quad + (1-K)^4 \cdot (3-2K)/48 \\
 &= (48-83K+80K^2-45K^3+14K^4-2K^5) / 48 .
 \end{aligned}$$

Portanto, as probabilidades de homozigose e heterozigose para o indivíduo III.1 (progênie normal do casal de primos) são, respectivamente:

$$\begin{aligned}
 P(\text{III.1} = n1, aa) &= [P_{(1)} + P_{(2)} + P_{(3)} + P_{(4)} + P_{(5)} + P_{(6)}] / \sum P_{(i)} \\
 &= \{ [12+12(1-K)+9(1-K)^2+3(1-K)^3 \\
 &\quad + (1-K)^4]/48 \} / P_{3K} \\
 &= (37-43K+24K^2-7K^3+K^4) \\
 &\quad / (48-83K+80K^2-45K^3+14K^4-2K^5)
 \end{aligned}$$

e

$$\begin{aligned}
 P(\text{III.1} = n1, Aa) &= [P_{(7)} + P_{(8)} + P_{(9)}] / \sum P_{(i)} \\
 &= \{ [6(1-K)^3+3(1-K)^4+2(1-K)^5]/48 \} / P_{3K} \\
 &= (11-40K+56K^2-38K^3+13K^4-2K^5) \\
 &\quad / (48-83K+80K^2-45K^3+14K^4-2K^5) .
 \end{aligned}$$

As deduções das fórmulas da probabilidade de homozigose e heterozigose para a progênie de outros tipos de uniões consanguíneas seguem o mesmo procedimento detalhado acima.

As tabelas abaixo mostram as funções de verossimilhança ($C = A + B$) da árvore e as quantidades **A** e **B** necessárias para a composição das probabilidades de heterozigose $P(het, nl) = A/C$ e de homozigose $P(hom, nl) = B/C$ para a progênie normal das uniões consanguíneas **1 a 11** (**1** - incesto entre irmãos; **2** - união tio-sobrinha ou tia-sobrinho; **3 a 11** - casais de primos em primeiro a nono graus (números ímpares: primeiro, terceiro, quinto, sétimo e nono graus; números pares: segundo, quarto, sexto e oitavo graus). Rogatko (1983) e Rogatko et al. (1986) calcularam essas probabilidades até o caso de primos em terceiro grau e forneceram um algoritmo geral para a dedução das fórmulas. Como algumas das fórmulas apresentadas nesses dois trabalhos apresentam erros algébricos ou tipográficos, recalculamos todas elas.

Tabela 12 - Quantidades necessárias (A) para a composição das probabilidades de heterozigose da progênie normal [P(het,nl)], considerando os tipos de união consanguínea descritos acima.

UNIÃO	A
1	$5 - 12K + 9K^2 - 2K^3$
2	$8 - 23K + 24K^2 - 11K^3 + 2K^4$
3	$11 - 40K + 56K^2 - 38K^3 + 13K^4 - 2K^5$
4	$17 - 69K + 114K^2 - 100K^3 + 51K^4 - 15K^5 + 2K^6$
5	$23 - 110K + 219K^2 - 238K^3 + 157K^4 - 66K^5 + 17K^6 - 2K^7$
6	$35 - 181K + 401K^2 - 505K^3 + 407K^4 - 223K^5 + 83K^6 - 19K^7 + 2K^8$
7	$47 - 276K + 702K^2 - 1026K^3 + 972K^4 - 642K^5 + 306K^6 - 102K^7 + 21K^8 - 2K^9$
8	$71 - 443K + 1218K^2 - 1968K^3 + 2118K^4 - 1638K^5 + 948K^6 - 408K^7 + 123K^8 - 23K^9 + 2K^{10}$
9	$95 - 658K + 2021K^2 - 3666K^3 + 4446K^4 - 3900K^5 + 2610K^6 - 1356K^7 + 531K^8 - 146K^9 + 25K^{10} - 2K^{11}$
10	$143 - 1041K + 3399K^2 - 6647K^3 + 8832K^4 - 8634K^5 + 6558K^6 - 3966K^7 + 1887K^8 - 677K^9 + 171K^{10} - 27K^{11} + 2K^{12}$
11	$191 - 1520K + 5448K^2 - 11726K^3 + 17159K^4 - 18474K^5 + 15528K^6 - 10572K^7 + 5853K^8 - 2564K^9 + 848K^{10} - 198K^{11} + 29K^{12} - 2K^{13}$

Tabela 13 - Quantidades necessárias (B) para a composição da probabilidade de homozigose da progênie normal $[P(\text{hom}, n1)]$, considerando os tipos de união consanguínea descritos acima.

UNIÃO	B
1	$7 - 5K + K^2$
2	$16 - 15K + 6K^2 - K^3$
3	$37 - 43K + 24K^2 - 7K^3 + K^4$
4	$79 - 104K + 73K^2 - 31K^3 + 8K^4 - K^5$
5	$169 - 249K + 207K^2 - 110K^3 + 39K^4 - 9K^5 + K^6$
6	$349 - 550K + 516K^2 - 329K^3 + 149K^4 - 48K^5 + 10K^6 - K^7$
7	$721 - 1211K + 1258K^2 - 917K^3 + 490K^4 - 197K^5 + 58K^6 - 11K^7 + K^8$
8	$1465 - 2556K + 2853K^2 - 2319K^3 + 1431K^4 - 687K^5 + 255K^6 - 69K^7 + 12K^8 - K^9$
9	$2977 - 5389K + 6417K^2 - 5700K^3 + 3918K^4 - 2142K^5 + 942K^6 - 324K^7 + 81K^8 - 13K^9 + K^{10}$
10	$6001 - 11102K + 13822K^2 - 13173K^3 + 9954K^4 - 6108K^5 + 3084K^6 - 1266K^7 + 405K^8 - 94K^9 + 14K^{10} - K^{11}$
11	$12097 - 22863K + 29676K^2 - 30067K^3 + 24519K^4 - 16446K^5 + 9240K^6 - 43506K^7 + 1671K^8 - 499K^9 + 108K^{10} - 15K^{11} + K^{12}$

Tabela 14 - Funções de verossimilhança (C) das árvores com uniões consanguíneas, segundo os tipos de união descritos acima.

UNIÃO	C
1	$12 - 17K + 10K^2 - 2K^3$
2	$24 - 38K + 30K^2 - 12K^3 + 2K^4$
3	$48 - 83K + 80K^2 - 45K^3 + 14K^4 - 2K^5$
4	$96 - 173K + 187K^2 - 131K^3 + 59K^4 - 16K^5 + 2K^6$
5	$192 - 359K + 426K^2 - 348K^3 + 196K^4 - 75K^5 + 18K^6 - 2K^7$
6	$384 - 731K + 917K^2 - 834K^3 + 556K^4 - 271K^5 + 93K^6 - 20K^7 + 2K^8$
7	$768 - 1487K + 1960K^2 - 1943K^3 + 1462K^4 - 839K^5 + 364K^6 - 113K^7 + 22K^8 - 2K^9$
8	$1536 - 2999K + 4071K^2 - 4287K^3 + 3549K^4 - 2325K^5 + 1203K^6 - 477K^7 + 135K^8 - 24K^9 + 2K^{10}$
9	$3072 - 6047K + 8438K^2 - 9366K^3 + 8364K^4 - 6042K^5 + 3552K^6 - 1680K^7 + 612K^8 - 159K^9 + 26K^{10} - 2K^{11}$
10	$6144 - 12143K + 17221K^2 - 19820K^3 + 18786K^4 - 14742K^5 + 9642K^6 - 5232K^7 + 2292K^8 - 771K^9 + 185K^{10} - 28K^{11} + 2K^{12}$
11	$12288 - 24383K + 35124K^2 - 41793K^3 + 41678K^4 - 34920K^5 + 24768K^6 - 14922K^7 + 7524K^8 - 3063K^9 + 956K^{10} - 213K^{11} + 30K^{12} - 2K^{13}$

Não existe interesse em obter-se uma fórmula geral para as expressões acima, uma vez que todas se referem apenas ao caso de estruturas consanguíneas sem ramificação e com apenas uma progênie do casal consanguíneo. Como no caso anterior, não existem soluções gerais em fórmula analítica simples para uma árvore consanguínea genérica qualquer e o assunto reveste-se apenas de curiosidade teórica, não havendo maiores dificuldades em determinar-se as quantidades acima referentes a uma árvore consanguínea qualquer através da aplicação de princípios de cálculo bayesiano.

IV) EXEMPLOS DE APLICAÇÃO DO MÉTODO

Apresentamos, neste capítulo, exemplos de aplicação do método visando a estimativa da taxa de penetrância (K) a partir de genealogias com afetados pela síndrome da ectrodactilia associada à hemimelia tibial.

Localizamos na literatura oito trabalhos descrevendo 21 famílias com afetados. Uma delas, apresentada parcialmente por Richieri-Costa et al. (1987), foi estudada no Laboratório de Genética Humana (LGH) do Departamento de Genética e Biologia Evolutiva do Instituto de Biociências da USP; empregando os dados dessa genealogia, detalhamos alguns modelos com taxa de penetrância constante e variável. Para as demais 20 famílias, apresentamos as estimativas de K constante e variável, apenas com dois valores de K, com discussões pertinentes a cada caso.

1) ESTUDO DE MODELOS COM TAXA DE PENETRÂNCIA CONSTANTE E VARIÁVEL

1.1) MODELO COM TAXA DE PENETRÂNCIA CONSTANTE (MODELO I)

No heredograma da família referida acima,

podemos identificar sete tipos básicos de configurações:

que representam, respectivamente:

- (1) indivíduos afetados (penetrantes);
- (2) indivíduos heterozigotos certos não-penetrantes;
- (3) a (6) - árvores de indivíduos normais descendentes de portadores certos do gene (penetrantes ou não), sendo uma árvore com apenas uma geração (3) e três árvores (4) a (6) com duas gerações;
- (7) gêmeos de mesmo sexo, normais, descendentes de portador certo do gene.

As probabilidades associadas a cada uma dessas configurações são:

- (1) K
- (2) $1-K$
- (3) $1/2 + (1-K)/2 = (2-K)/2$
- (4) $1/2 + (1-K)/2 \cdot [1/2 + (1-K)/2] = [2 + (1-K)(2-K)]/4$
- (5) $1/2 + (1-K)/2 \cdot [1/2 + (1-K)/2]^2 = [4 + (1-K)(2-K)^2]/8$
- (6) $1/2 + (1-K)/2 \cdot [1/2 + (1-K)/2]^3 = [8 + (1-K)(2-K)^3]/16$
- (7) $1/3 \cdot [1/2 + (1-K)^2/2] + 2/3 \cdot [(2-K)/2]^2 = (3-3K+K^2)/3$.

Dado que as configurações (1) a (7) ocorrem, respectivamente, com as frequências absolutas 12, 11, 14, 2, 4, 3 e 1, a função de verossimilhança P_K de ocorrência da genealogia, considerando o valor de penetrância K , é

$$P_K = K^{12} \cdot (1-K)^{11} \cdot (2-K)^{14} \cdot [2+(1-K)(2-K)]^2 \cdot [4+(1-K)(2-K)]^4 \cdot [8+(1-K)(2-K)^3]^3 \cdot (3-3K+K^2)$$

A estimativa do valor da taxa de penetrância K e os gráficos da função sob forma logarítmica $[\log(P_K)]$ e sua derivada (dL/dK) podem ser obtidos, por exemplo, através do seguinte código Mathematica:

```
L = 12*Log[K] + 11*Log[1-K] + 14*Log[2-K] + 2*Log[2+(1-K)*(2-K)] +
    4*Log[4+(1-K)*(2-K)^2] + 3*Log[8+(1-K)*(2-K)^3] + Log[(3-3K+K^2)];
dLdK = D[L,K];
Plot[{L/100,dLdK/100},{K,0,1}, PlotRange->{1,-1},
    AspectRatio->2, Frame->True]
FindRoot[dLdK==0,{K,0.5}]
```

```
{K -> 0.351665}
```

O intervalo exato de confiança a 95% da estimativa ($K = 0,352$) é $0,200 - 0,541$. Esses limites foram calculados pelo programa **CONFINT3.BAS (ANEXO I)**.

O gráfico, gerado pelo seguinte código Mathematica, mostra a função P_K , a estimativa $K = 0,352$ e os limites inferior e superior do seu intervalo exato de confiança a 95%:

```
a=0.200;
b=0.541;
c=0.352;
fa= a^12*(1-a)^11*(2-a)^14*(3-3*a+a^2)*(2+(1-a)*(2-a))^2*
    (4+(1-a)*(2-a)^2)^4*(8+(1-a)*(2-a)^3)^3;
fb= b^12*(1-b)^11*(2-b)^14*(3-3*b+b^2)*(2+(1-b)*(2-b))^2*
    (4+(1-b)*(2-b)^2)^4*(8+(1-b)*(2-b)^3)^3;
fc= c^12*(1-c)^11*(2-c)^14*(3-3*c+c^2)*(2+(1-c)*(2-c))^2*
    (4+(1-c)*(2-c)^2)^4*(8+(1-c)*(2-c)^3)^3;
Show[
  Plot[k^12*(1-k)^11*(2-k)^14*(3-3*k+k^2)*(2+(1-k)*(2-k))^2*
    (4+(1-k)*(2-k)^2)^4*(8+(1-k)*(2-k)^3)^3, {k,0,1},
    Frame->True,
    DisplayFunction -> Identity],
  Graphics[{
    Line[{{a,0},{a,fa}}],
    Line[{{c,0},{c,fc}}],
    Line[{{b,0},{b,fb}}],
    }],PlotRange->{0,320},
  DisplayFunction -> $DisplayFunction,
  AspectRatio -> 1/2];
```

As probabilidades de homozigose e heterozigose de cada indivíduo inserido nas árvores normais (3) a (7), referidas anteriormente, são, respectivamente:

$$P(aa) = 1/(2-K)$$

$$P(Aa) = (1-K)/(2-K)$$

$$P_1(aa) = 2/[2+(1-K)(2-K)]$$

$$P_1(Aa) = (1-K)(2-K)/[2+(1-K)(2-K)]$$

$$P_2(aa) = (3-K)/[2+(1-K)(2-K)]$$

$$P_2(Aa) = (1-K)^2/[2+(1-K)(2-K)]$$

$$P_3(aa) = 4/[4+(1-K)(2-K)^2]$$

$$P_3(Aa) = (1-K)(2-K)^2/[4+(1-K)(2-K)^2]$$

$$P_4(aa) = [4+(1-K)(2-K)]/[4+(1-K)(2-K)^2]$$

$$P_4(Aa) = (1-K)^2(2-K)/[4+(1-K)(2-K)^2]$$

$$P_5(aa) = P_4(aa)$$

$$P_5(Aa) = P_4(Aa)$$

$$\begin{aligned}
 P_6(aa) &= 8 / [8 + (1-K)(2-K)^3] \\
 P_6(Aa) &= (1-K)(2-K)^3 / [8 + (1-K)(2-K)^3] \\
 P_7(aa) &= [8 + (1-K)(2-K)^2] / [8 + (1-K)(2-K)^3] \\
 P_7(Aa) &= (1-K)^2(2-K)^2 / [8 + (1-K)(2-K)^3] \\
 P_8(aa) &= P_9(aa) = P_7(aa) \\
 P_8(Aa) &= P_9(Aa) = P_7(Aa)
 \end{aligned}$$

$$\begin{aligned}
 P_{10}(aa) &= (3-K) / (6-6K+2K^2) \\
 P_{10}(Aa) &= (3-5K+2K^2) / (6-6K+2K^2) \\
 P_{11}(aa) &= P_{10}(aa) \\
 P_{11}(Aa) &= P_{10}(Aa)
 \end{aligned}$$

Na Tabela 15, apresentamos os valores das probabilidades de heterozigose $[P(\text{het})]$, para o caso de $K = 0,352$. As probabilidades de homozigose $[P(\text{hom})]$ são dadas por $P(\text{hom}) = 1 - P(\text{het})$.

Tabela 15 - Probabilidade de heterozigose $[P(\text{het})]$, sob modelo com taxa de penetrância constante ($K = 0,352$).

INDIVÍDUOS	P (HET)
II.2-3, IV.3, IV.9-11, V.1-2, V.4-5, V.8-11	0,393
III.6, IV.20	0,348
IV.18, V.19	0,137
II.5, IV.17, IV.19, IV.21	0,306
III.7-8, V.15-16, V.17-18, V.20-21	0,120
III.3, III.10, IV.16	0,266
IV.12-14, IV.23-25, V.12-14	0,105

IV.4-5	0,360
--------	-------

Os riscos de progênie afetada para cada um desses indivíduos são obtidos multiplicando-se os valores de **P (het)** por **K/2**.

1.2) MODELOS COM TAXA DE PENETRÂNCIA VARIÁVEL

A inspeção da genealogia em estudo sugere que o valor da taxa de penetrância seja variável com o tempo.

1.2.1) MODELO COM UM VALOR DE K POR GERAÇÃO (MODELO IV)

Agregando o indivíduo **I.1** à segunda geração, reduzimos o número de gerações da genealogia de cinco para quatro. Dessa forma, os números de indivíduos heterozigotos certos afetados e normais encontrados, respectivamente, na primeira, segunda, terceira e quarta gerações são:

Heterozigotos certos	Geração			
	1	2	3	4
Afetados	1	1	7	3
Normais	5	5	1	0

Sendo K_t e K_{t+1} os valores da taxa de penetrância em duas gerações sucessivas, as probabilidades associadas com as configurações (1) a (7) descritas anteriormente são, num modelo que assume o valor de penetrância variável **K** através das gerações:

(1) K_t

(2) $1-K_t$

$$\begin{aligned}
(3) \quad & 1/2 + (1-K_t)/2 = (2-K_t)/2 \\
(4) \quad & 1/2 + (1-K_t)/2 \cdot [1/2 + (1-K_{t+1})/2] = [2 + (1-K_t)(2-K_{t+1})]/4 \\
(5) \quad & 1/2 + (1-K_t)/2 \cdot [1/2 + (1-K_{t+1})/2]^2 = [4 + (1-K_t)(2-K_{t+1})^2]/8 \\
(6) \quad & 1/2 + (1-K_t)/2 \cdot [1/2 + (1-K_{t+1})/2]^3 = [8 + (1-K_t)(2-K_{t+1})^3]/16 \\
(7) \quad & 1/3 \cdot [1/2 + (1-K_t)/2]^2/2 + 2/3 \cdot [(2-K_t)/2]^2 = (3 - 3K_t + K_t^2)/3 .
\end{aligned}$$

As probabilidades de homozigose e heterozigose para cada indivíduo dentro das árvores de indivíduos normais tomam os seguintes valores:

$$\begin{aligned}
P(aa) &= 1/(2-K_t) \\
P(Aa) &= (1-K_t)/(2-K_t)
\end{aligned}$$

$$\begin{aligned}
P_1(aa) &= 2/[2 + (1-K_t)(2-K_{t+1})] \\
P_1(Aa) &= (1-K_t)(2-K_{t+1})/[2 + (1-K_t)(2-K_{t+1})] \\
P_2(aa) &= [3-K_t]/[2 + (1-K_t)(2-K_{t+1})] \\
P_2(Aa) &= (1-K_t)(1-K_{t+1})/[2 + (1-K_t)(2-K_{t+1})]
\end{aligned}$$

$$\begin{aligned}
P_3(aa) &= 4/[4 + (1-K_t)(2-K_{t+1})^2] \\
P_3(Aa) &= (1-K_t)(2-K_{t+1})^2/[4 + (1-K_t)(2-K_{t+1})^2] \\
P_4(aa) &= [4 + (1-K_t)(2-K_{t+1})]/[4 + (1-K_t)(2-K_{t+1})^2] \\
P_4(Aa) &= (1-K_t)(1-K_{t+1})(2-K_{t+1})/[4 + (1-K_t)(2-K_{t+1})^2] \\
P_5(aa) &= P_4(aa) \\
P_5(Aa) &= P_4(Aa)
\end{aligned}$$

$$\begin{aligned}
P_6(aa) &= 8/[8 + (1-K_t)(2-K_{t+1})^3] \\
P_6(Aa) &= (1-K_t)(2-K_{t+1})^3/[8 + (1-K_t)(2-K_{t+1})^3] \\
P_7(aa) &= [8 + (1-K_t)(2-K_{t+1})^2]/[8 + (1-K_t)(2-K_{t+1})^3] \\
P_7(Aa) &= (1-K_t)(1-K_{t+1})(2-K_{t+1})^2/[8 + (1-K_t)(2-K_{t+1})^3]
\end{aligned}$$

$$P_8(aa) = P_9(aa) = P_7(aa)$$

$$P_8(Aa) = P_9(Aa) = P_7(Aa)$$

$$P_{10}(aa) = (3-K_t) / (6-6K_t+2K_t^2)$$

$$P_{10}(Aa) = (3-5K_t+2K_t^2) / (6-6K_t+2K_t^2)$$

$$P_{11}(aa) = P_{10}(aa)$$

$$P_{11}(Aa) = P_{10}(Aa)$$

Considerando o número de vezes com que cada configuração (1) a (7) ocorre na genealogia e a geração (ou pares de gerações) a que pertence, obtemos a função de verossimilhança

$$\begin{aligned} P_K = & K_1 \cdot K_2 \cdot K_3^7 \cdot K_4^3 \cdot (1-K_1)^5 \cdot (1-K_2)^5 \cdot (1-K_3) \cdot (2-K_1)^2 \cdot (2-K_3)^4 \cdot (2-K_4)^8 \\ & \cdot [2+(1-K_2)(2-K_3)] \cdot [2+(1-K_3)(2-K_4)] \cdot [4+(1-K_1)(2-K_2)^2] \\ & \cdot [4+(1-K_3)(2-K_4)^2]^3 \cdot [8+(1-K_2)(2-K_3)^3]^2 \cdot [8+(1-K_3)(2-K_4)^3] \\ & \cdot (3-3K_3+K_3^2) \quad , \end{aligned}$$

onde K_1 , K_2 , K_3 e K_4 representam as taxas de penetrância na primeira, segunda, terceira e quarta gerações, respectivamente.

As derivadas primeiras (derivadas parciais) $\partial L / \partial K_1$, $\partial L / \partial K_2$, $\partial L / \partial K_3$ e $\partial L / \partial K_4$ da função $L = \text{Log}(P_K)$ assumem os seguintes valores literais:

$$\begin{aligned} \partial L / \partial K_1 &= -5 / (1-K_1) - 2 / (2-K_1) + 1 / K_1 - (2-K_2)^2 / [4+(1-K_1)(2-K_2)^2] \\ \partial L / \partial K_2 &= 1 / K_2 - 5 / (1-K_2) - (2-K_3) / [2+(1-K_2)(2-K_3)] \\ &\quad - 2(1-K_1)(2-K_2) / [4+(1-K_1)(2-K_2)^2] - 2(2-K_3)^3 / [8+(1-K_2)(2-K_3)^3] \\ \partial L / \partial K_3 &= 7 / K_3 - 1 / (1-K_3) - 4 / (2-K_3) - (1-K_2) / [2+(1-K_2)(2-K_3)] \\ &\quad - (2-K_4) / [2+(1-K_3)(2-K_4)] - 3(2-K_4)^2 / [4+(1-K_3)(2-K_4)^2] \\ &\quad - 6(1-K_2)(2-K_3)^2 / [8+(1-K_2)(2-K_3)^3] - (2-K_4)^3 / [8+(1-K_3)(2-K_4)^3] \\ &\quad - (3-2K_3) / (3-3K_3+K_3^2) \\ \partial L / \partial K_4 &= 3 / K_4 - 8 / (2-K_4) - (1-K_3) / [2+(1-K_3)(2-K_4)] \end{aligned}$$

$$\begin{aligned}
& - 6(1-K_3)(2-K_4)/[4+(1-K_3)(2-K_4)^2] \\
& - 3(1-K_3)(2-K_4)^2/[8+(1-K_3)(2-K_4)^3]
\end{aligned}$$

As derivadas segundas $\partial L(\partial L/\partial K_1)/\partial K_1 = \partial^2 L/\partial K_1^2$, etc.

são:

$$\begin{aligned}
\partial^2 L/\partial K_1^2 &= -5/(1-K_1)^2 - 2/(2-K_1)^2 - 1/K_1^2 - (2-K_2)^4/[4+(1-K_1)(2-K_2)^2]^2 \\
\partial^2 L/\partial K_1 K_2 &= 2(2-K_2)/[4+(1-K_1)(2-K_2)^2] - 2(1-K_1)(2-K_2)^3/[4+(1-K_1)(2-K_2)^2]^2 \\
\partial^2 L/\partial K_1 K_3 &= \partial^2 L/\partial K_1 K_4 = 0 \\
\partial^2 L/\partial K_2^2 &= 2(1-K_1)/[4+(1-K_1)(2-K_2)^2] - 5/(1-K_2)^2 - 1/K_2^2 \\
&\quad - 4(1-K_1)^2(2-K_2)^2/[4+(1-K_1)(2-K_2)^2]^2 \\
&\quad - 2(2-K_3)^6/[8+(1-K_2)(2-K_3)^3]^2 \\
&\quad + (2-K_3)(-2+K_3)/[2+(1-K_2)(2-K_3)]^2 \\
\partial^2 L/\partial K_2 K_1 &= \partial^2 L/\partial K_1 K_2 \\
\partial^2 L/\partial K_2 K_3 &= 1/[2+(1-K_2)(2-K_3)] - (1-K_2)(2-K_3)/[2+(1-K_2)(2-K_3)]^2 \\
&\quad + 6(2-K_3)^2/[8+(1-K_2)(2-K_3)^3] \\
&\quad - 6(1-K_2)(2-K_3)^5/[8+(1-K_2)(2-K_3)^3]^2 \\
\partial^2 L/\partial K_2 K_4 &= 0 \\
\partial^2 L/\partial K_3^2 &= (1-K_2)(-1+K_2)/[2+(1-K_2)(2-K_3)]^2 - 1/(1-K_3)^2 - 4/(2-K_3)^2 \\
&\quad + 12(1-K_2)(2-K_3)/[8+(1-K_2)(2-K_3)^3] \\
&\quad - 18(1-K_2)^2(2-K_3)^4/[8+(1-K_2)(2-K_3)^3]^2 \\
&\quad - 7/K_3^2 - 3(2-K_4)^4/[4+(1-K_3)(2-K_4)^2]^2 \\
&\quad - (2-K_4)^6/[8+(1-K_3)(2-K_4)^3]^2 - (2-K_4)(-2+K_4)/[2+(1-K_3)(2-K_4)]^2 \\
&\quad + (3-2K_3)(-3+2K_3)/(3-3K_3+K_3^2)^2 + 2/(3-3K_3+K_3^2) \\
\partial^2 L/\partial K_3 K_1 &= 0 \\
\partial^2 L/\partial K_3 K_2 &= \partial^2 L/\partial K_2 K_3 \\
\partial^2 L/\partial K_3 K_4 &= 1/[2+(1-K_3)(2-K_4)] - (1-K_3)(2-K_4)/[(2+(1-K_3)(2-K_4)]^2 \\
&\quad + 6(2-K_4)/[4+(1-K_3)(2-K_4)^2] + 3(2-K_4)^2/[8+(1-K_3)(2-K_4)^3] \\
&\quad - 6(1-K_3)(2-K_4)^3/[4+(1-K_3)(2-K_4)^2]^2
\end{aligned}$$

$$\begin{aligned}
& - 3 (1-K_3) (2-K_4)^5 / [8 + (1-K_3) (2-K_4)^3]^2 \\
\partial^2 L / \partial K_4^2 &= (1-K_3) (-1+K_3) / [2 + (1-K_3) (2-K_4)]^2 + 6 (1-K_3) / [4 + (1-K_3) (2-K_4)^2] \\
& - 8 / (2-K_4)^2 + 6 (1-K_3) (2-K_4) / [8 + (1-K_3) (2-K_4)^3] \\
& - 12 (1-K_3)^2 (2-K_4)^2 / [4 + (1-K_3) (2-K_4)^2]^2 \\
& - 9 (1-K_3)^2 (2-K_4)^4 / [8 + (1-K_3) (2-K_4)^3]^2 - 3 / K_4^2 \\
\partial^2 L / \partial K_4 K_1 &= \partial^2 L / \partial K_4 K_2 = 0 \\
\partial^2 L / \partial K_4 K_3 &= \partial^2 L / \partial K_3 K_4
\end{aligned}$$

As estimativas de K_1 , K_2 , K_3 e K_4 são obtidas dessas quantidades empregando, por exemplo, o programa **KVAR_4.BAS (ANEXO I)**, o qual apresenta os seguintes resultados:

i	K1	K2	K3	K4	dL/dK1	dL/dK2	dL/dK3	dL/dK4
0	0.99000	0.99000	0.01000	0.01000	-501.22495	-501.93154	693.44226	293.49382
1	0.97998	0.97996	0.01991	0.01978	-250.89993	-251.39962	345.04879	145.13689
2	0.95986	0.95979	0.03943	0.03871	-125.71261	-126.07288	170.80785	70.97462
3	0.91938	0.91909	0.07731	0.07413	-63.06967	-63.30131	83.60712	33.92919
4	0.83766	0.83678	0.14832	0.13610	-31.64766	-31.73688	39.87219	15.48844
5	0.67301	0.67142	0.27122	0.23078	-15.69813	-15.65609	17.80750	6.45568
6	0.35749	0.35800	0.44611	0.34269	-6.67266	-6.60227	6.57814	2.30606
7	0.03208	0.03738	0.60104	0.42952	24.49438	20.13257	1.28789	0.67522
8	0.05713	0.06537	0.65045	0.46779	10.67418	8.57198	0.05032	0.12146
9	0.09129	0.10114	0.65402	0.47536	3.90858	2.95256	0.00880	0.00675
10	0.12217	0.12957	0.65548	0.47587	0.92984	0.60438	-0.00009	0.00214
11	0.13472	0.13873	0.65585	0.47603	0.07714	0.03542	-0.00001	0.00053
12	0.13595	0.13934	0.65587	0.47606	0.00061	0.00014	0.00003	0.00004
13	0.13596	0.13934	0.65587	0.47607	0.00000	0.00000	0.00000	0.00000

VARIANCE-COVARIANCE MATRIX var(i, j), i, j = 1, ..., 4

	1	2	3	4
1	0.01623073	0.00008453	0.00000337	0.00000023
2	0.00008453	0.01710250	0.00068099	0.00004638
3	0.00000337	0.00068099	0.03785926	0.00257845
4	0.00000023	0.00004638	0.00257845	0.06213517

APPR. 95% CONFID. INTERVALS OF K_i ESTIMATES

i	K_i	ll(K_i) - ul(K_i)
1	0.13596	0.0000 - 0.3857
2	0.13934	0.0000 - 0.3957
3	0.65587	0.2745 - 1.0000
4	0.47607	0.0000 - 0.9646

As estimativas $K_1 = 0,136$, $K_2 = 0,139$, $K_3 = 0,656$ e $K_4 = 0,476$ são sempre obtidas, independentemente dos valores iniciais informados ao programa, como no exemplo numérico mostrado acima. Como a amostra da qual as estimativas foram obtidas é pequena, as quantidades contidas na matriz de variância-covariância são relativamente grandes, bem como o intervalo de confiança a 95% dessas estimativas.

Os valores numéricos das probabilidades de homozigose e heterozigose para cada indivíduo dentro das árvores de indivíduos normais (3) a (7) dependem das gerações ou pares de gerações nas quais os indivíduos estão incluídos:

Configuração	Ocorrências por geração			
	1	2	3	4
(3)	2	0	4	8
(4)	0		1	1
(5)	1		0	3
(6)	0		2	1
(7)			1	

Usando as estimativas de K_i , apresentamos na Tabela 16 os valores de probabilidade de heterozigose $[P(het)]$ de todos os indivíduos dentro das árvores de indivíduos normais (3) a (7).

Tabela 16 - Probabilidades de heterozigose [P(het)], sob modelo com taxa de penetrância variável por geração ($K_1 = 0,136$, $K_2 = 0,139$, $K_3 = 0,656$ e $K_4 = 0,476$).

INDIVÍDUOS	P (HET)
II.2, II.3	0,463
IV.3, IV.9-11	0,256
V.1-2, V.4-5, V.8-11	0,344
III.6	0,366
IV.18	0,094
IV.20	0,208
V.19	0,071
II.5	0,428
III.7-8	0,198
IV.17, IV.19, IV.21	0,166
V.15-16, V.17-18, V.20-21	0,057
III.3, III.10	0,207
IV.12-14, IV.23-25	0,053
IV.16	0,132
V.12-14	0,045
IV.4-5	0,199

1.2.2) MODELO COM DOIS VALORES DE K (MODELO II)

O modelo apresentado na seção anterior pode ser simplificado reunindo-se as gerações em dois grupos, de forma a obter duas estimativas da taxa de penetrância (K_1 e K_2).

Agrupamos as cinco gerações de quatro formas distintas, buscando definir a melhor maneira de reuni-las:

- (1) primeira geração com K_1 e demais com K_2 ;
- (2) duas primeiras gerações com K_1 e demais com K_2 ;
- (3) três primeiras gerações com K_1 e demais com K_2 ;
- (4) quatro primeiras gerações com K_1 e última com K_2 .

As funções de verossimilhança e as estimativas das taxas de penetrância K_1 e K_2 , obtidas em cada caso, são contrastadas com as obtidas através do modelo com K constante, empregando-se o teste de razão de verossimilhanças (teste G), sendo $G = -2 \cdot [\log(P_K) - \log(P_{K_1, K_2})]$. Apresentamos na Tabela 17, as estimativas de K_1 e K_2 e valor do teste G em cada caso.

Tabela 17 - Estimativas das taxas de penetrância K_1 e K_2 e valor do teste G contrastando os modelos com duas estimativas de K e K constante ($K = 0,352$).

Agrupamento	K_1	K_2	$G_{(1 \text{ g.l.})}$	P
1	0	0,365	0,887	0,346
2	0,137	0,425	2,111	0,146
3	0,137	0,600	6,110	0,013
4	0,335	0,435	0,177	0,674

Apenas as estimativas obtidas por meio do agrupamento (3) revelaram-se melhores do que a obtida com o modelo com K constante. Detalhamos, a seguir, como essas estimativas foram obtidas.

Reunindo num grupo os dados das gerações 1, 2 e 3 e noutro os das gerações 4 e 5, obtemos a função de verossimilhança

$$P_K = K_1^2 \cdot K_2^{10} \cdot (1-K_1)^{10} \cdot (1-K_2) \cdot (2-K_1)^2 \cdot (2-K_2)^{12} \cdot [2+(1-K_1)(2-K_2)] \\ \cdot [2+(1-K_2)(2-K_2)] \cdot [4+(1-K_1)(2-K_1)^2] \cdot [4+(1-K_2)(2-K_2)^2]^3 \\ \cdot [8+(1-K_1)(2-K_2)^3]^2 \cdot [8+(1-K_2)(2-K_2)^3] \cdot (3-3K_2+K_2^2) \quad .$$

As derivadas parciais $\partial L / \partial K_1$ e $\partial L / \partial K_2$ dessa função tomam os seguintes valores literais:

$$\begin{aligned} \partial L / \partial K_1 &= 2/K_1 - 10/(1-K_1) - 2/(2-K_1) - (2-K_2)/[2+(1-K_1)(2-K_2)] \\ &\quad - 2(1-K_1)(2-K_1) - (2-K_1)^2/[4+(1-K_1)(2-K_1)^2] \\ &\quad - 2(2-K_2)^3/[8+(1-K_1)(2-K_2)^3] \\ \partial L / \partial K_2 &= 10/K_2 - 1/(1-K_2) - 12/(2-K_2) - (1-K_1)/[2+(1-K_1)(2-K_2)] \\ &\quad - (3-2K_2)/[2+(1-K_2)(2-K_2)] \\ &\quad + 3(-2(1-K_2)(2-K_2) - (2-K_2)^2)/[4+(1-K_2)(2-K_2)^2] \\ &\quad - 6(1-K_1)(2-K_2)^2/[8+(1-K_1)(2-K_2)^3] \\ &\quad + [-3(1-K_2)(2-K_2)^2 - (2-K_2)^3]/[8+(1-K_2)(2-K_2)^3] \\ &\quad - (3-2K_2)/(3-3K_2+K_2^2) \end{aligned}$$

As derivadas segundas, obtidas de

$$\begin{aligned} \partial(\partial L / \partial K_1) / \partial K_1 &= \partial^2 L / \partial K_1^2 \\ \partial(\partial L / \partial K_1) / \partial K_2 &= \partial(\partial L / \partial K_2) / \partial K_1 = \partial^2 L / \partial K_1 \partial K_2 \\ \partial(\partial L / \partial K_2) / \partial K_2 &= \partial^2 L / \partial K_2^2, \text{ são:} \end{aligned}$$

$$\begin{aligned} \partial^2 L / \partial K_1^2 &= [2(1-K_1)+4(2-K_1)]/[4+(1-K_1)(2-K_1)^2] - 10/(1-K_1)^2 \\ &\quad - 2/(2-K_1)^2 - [-2(1-K_1)(2-K_1) - (2-K_1)^2]/[4+(1-K_1)(2-K_1)^2]^2 - 2/K_1^2 \\ &\quad - 2(2-K_2)^6/[8+(1-K_1)(2-K_2)^3]^2 \\ &\quad - (2-K_2)(-2+K_2)/[2+(1-K_1)(2-K_2)]^2 \end{aligned}$$

$$\begin{aligned}
\partial^2 L / \partial K_1 K_2 &= \partial^2 L / dK_2 K_1 = 1 / [2 + (1 - K_1) (2 - K_2)] \\
&- (1 - K_1) (2 - K_2) / [2 + (1 - K_1) (2 - K_2)]^2 \\
&+ 6 (2 - K_2)^2 / [8 + (1 - K_1) (2 - K_2)^3] \\
&- 6 (1 - K_1) (2 - K_2)^5 / [8 + (1 - K_1) (2 - K_2)^3]^2 \\
\partial^2 L / \partial K_2^2 &= (1 - K_1) (-1 + K_1) / [2 + (1 - K_1) (2 - K_2)]^2 + 2 / [2 + (1 - K_2) (2 - K_2)] \\
&+ 3 [2 (1 - K_2) + 4 (2 - K_2)] / [4 + (1 - K_2) (2 - K_2)^2] - 1 / (1 - K_2)^2 \\
&- 12 / (2 - K_2)^2 + 12 (1 - K_1) (2 - K_2) / [8 + (1 - K_1) (2 - K_2)^3] \\
&+ [6 (1 - K_2) (2 - K_2) + 6 (2 - K_2)^2] / [8 + (1 - K_2) (2 - K_2)^3] \\
&- 3 [-2 (1 - K_2) (2 - K_2) - (2 - K_2)^2]^2 / [4 + (1 - K_2) (2 - K_2)^2]^2 \\
&- [-3 (1 - K_2) (2 - K_2)^2 - (2 - K_2)^3]^2 / [8 + (1 - K_2) (2 - K_2)^3]^2 \\
&- 18 (1 - K_1)^2 (2 - K_2)^4 / [8 + (1 - K_1) (2 - K_2)^3]^2 \\
&- 10 / K_2^2 + (3 - 2K_2) (-3 + 2K_2) / [2 + (1 - K_2) (2 - K_2)]^2 \\
&+ (3 - 2K_2) (-3 + 2K_2) / [3 - 3K_2 + K_2^2]^2 + 2 / [3 - 3K_2 + K_2^2]
\end{aligned}$$

As estimativas K_1 e K_2 foram obtidas empregando-se o programa **KVAR_2.BAS (ANEXO I)**. Como no caso anterior de quatro valores de taxa de penetrância, o modelo converge adequadamente para as estimativas K_1 e K_2 mesmo quando partimos de valores iniciais inapropriados, como no exemplo a seguir:

i	K1	K2	dL/dK1	dL/dK2	var (K1)	cov (K1, K2)	var (K2)
0	0.99000	0.01000	-1003.15649	986.93609	0.00001	0.00000	0.00001
1	0.97997	0.01987	-502.29959	490.17708	0.00004	0.00000	0.00004
2	0.95982	0.03921	-251.78588	241.76131	0.00016	0.00000	0.00015
3	0.91924	0.07636	-126.37274	117.49326	0.00065	0.00000	0.00058
4	0.83721	0.14461	-63.39035	55.27931	0.00262	0.00001	0.00208
5	0.67213	0.25875	-31.36829	24.12420	0.01014	0.00012	0.00658
6	0.35708	0.41353	-13.29036	8.70173	0.02470	0.00052	0.01613
7	0.03333	0.54704	46.66979	1.83979	0.00055	0.00001	0.02563
8	0.05911	0.59486	20.24488	0.09767	0.00171	0.00005	0.02838
9	0.09378	0.59864	7.31824	0.00076	0.00417	0.00012	0.02856

10	0.12426	0.59957	1.68772	0.00030	0.00699	0.00021	0.02860
11	0.13606	0.59994	0.13044	0.00005	0.00820	0.00025	0.02862
12	0.13713	0.59997	0.00088	0.00000	0.00831	0.00025	0.02862
13	0.13713	0.59997	0.00000	0.00000	0.00832	0.00025	0.02862

As probabilidades de heterozigose de todos os indivíduos incluídos nas árvores de indivíduos normais são apresentadas na Tabela abaixo.

Tabela 18 - Probabilidades de heterozigose [P(het)], sob modelo com taxa de penetrância variável por geração ($K_1 = 0,137$ e $K_2 = 0,600$).

INDIVÍDUOS	P (HET)
II.2-3	0,463
IV.3, IV.9-11, V.1-2, V.4-5, V.8-11	0,286
III.6	0,377
IV.18	0,108
IV.20	0,219
V.19	0,062
II.5	0,428
III.7-8	0,198
IV.17, IV.19, IV.21	0,164
V.15-16, V.17-18, V.20-21	0,047
III.3 , III.10	0,228
IV.12-14, IV.23-25	0,065
IV.16	0,121
V.12-14	0,034
IV.4-5	0,231

A Tabela 19 lista os valores das probabilidades de heterozigose de todos os indivíduos das árvores normais sob os três modelos estudados. Indivíduos heterozigotos certos, sejam afetados ou normais, mantém um valor de **P(het) = 1** sob quaisquer hipóteses.

Tabela 19 - Probabilidades de heterozigose considerando os modelos com taxa de penetrância constante (Modelo I) e variável (Modelos II, com duas estimativas de K e IV, com quatro estimativas de K).

INDIVÍDUOS	PROBABILIDADE DE HETEROZIGOSE		
	MODELO I	MODELO II	MODELO IV
II.2-3	0,393	0,463	0,463
IV.3, IV.9-11		0,286	0,256
V.1-2, V.4-5, V.8-11			0,344
III.6	0,348	0,377	0,366
IV.18	0,137	0,108	0,094
IV.20	0,348	0,219	0,208
V.19	0,137	0,062	0,071
II.5	0,306	0,428	0,428
III.7-8	0,120	0,198	0,198
IV.17, IV.19, IV.21	0,306	0,164	0,166
V.15-16, V.17-18, V.20-21	0,120	0,047	0,057
III.3, III.10	0,266	0,228	0,207
IV.12-14, IV.23-25	0,105	0,065	0,053
IV.16	0,266	0,121	0,132
V.12-14	0,105	0,034	0,045
IV.4-5	0,360	0,231	0,199

1.2.3) TESTE DE RAZÃO DE VEROSSIMILHANÇAS (TESTE G)

Realizamos dois testes de razão de verossimilhanças, com o objetivo de identificar o modelo (com K constante ou variável) mais adequado à genealogia. Os testes contrastaram os modelos:

- (1) duas estimativas de K contra K constante;
- (2) quatro estimativas de K contra duas estimativas de K.

Os valores dos testes (1) e (2) foram, respectivamente, **6,110** e **0,341**, correspondendo apenas o valor do primeiro teste a um resultado estatisticamente significativo ao nível de cerca de 1% (**P = 0,0134**). Esses resultados indicam que o modelo com duas estimativas de K (**K₁ e K₂**) explica a estrutura da genealogia de forma mais adequada que os demais.

O modelo com taxas variáveis de penetrância encontra uma importante aplicação nos programas que estimam taxas de ligação, os quais geralmente solicitam informação sobre a taxa de penetrância do gene envolvido na determinação de uma doença ou defeito presente numa genealogia qualquer estudada com marcadores. Obviamente a adoção de um modelo de penetrância coerente com o comportamento do traço em estudo na genealogia é de fundamental importância para a determinação de valores confiáveis das probabilidades favorecendo a hipótese de ligamento fatorial entre o traço em estudo e os marcadores empregados.

2) ANÁLISE DE GENEALOGIAS DESCRITAS NA LITERATURA

Na Tabela 20, apresentamos a identificação de cada uma das famílias referidas neste estudo.

Tabela 20 - Identificação das genealogias utilizadas na presente seção.

TRABALHOS	FAMÍLIA	GENEALOGIA
RICHIERI-COSTA et al. (1987)	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
HOYME et al. (1987)	1	7
	2	8
	3	9
	4	10
SPRANGER & SCHAPER (1988)	1	11
MAJEWSKI et al. (1985)	1	12
	2	13
	3	14
	4	15
	6	16
WITTERS et al. (2001)	1	17
SENER et al. (1989)	1	18
MARINONI et al. (1994)	1	19
NAVEED et al. (2006)	1	20
MAJEWSKI et al. (1996)	1	21

A genealogia **3** descrita por Richieri-Costa et al. (1987) é uma parte da família estudada em nosso laboratório, cujos resultados foram apresentados no item **1.1**.

Para as demais famílias, apresentamos em sequência:

- o heredograma filtrado;
- as funções de verossimilhança P_K (K constante) e P_{K_1, K_2} (K variável);
- a estimativa da taxa de penetrância constante (K) e seu respectivo intervalo exato de confiança a 95% $[IC95\%(K)]$;
- as estimativas da taxa de penetrância variável (K_1 e K_2);
- o valor do teste de razão de verossimilhanças $[G_{(1 \text{ g.l.})}]$. O valor da probabilidade P correspondente ao valor do teste G foi obtida utilizando-se o programa **Statistical Distributions** (Otto & Horimoto, 2006);
- o gráfico da função P_K sob a forma logarítmica $[\log(P_K)]$ e sua derivada (dL/dK) ;
- o gráfico da função P_K com o $IC95\%(K)$.

2.1) GENEALOGIA 1

$$P_K = K^5 \cdot (1-K)^3 \cdot (2-K)^{20}$$

$$P_{K_1, K_2} = K_1 \cdot K_2^4 \cdot (1-K_1)^2 \cdot (1-K_2) \cdot (2-K_1)^5 \cdot (2-K_2)^{15}$$

K constante		K variável		$G_{(1 \text{ g.l.})}$
K	IC95% (K)	K_1	K_2	0,497
0,309	0,123 - 0,578	0,191	0,371	(P = 0,481)

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.2) GENEALOGIA 2

$$P_K = K^7 \cdot (1-K)^3 \cdot (2-K)^5 \cdot [2 + (1-K)(2-K)] \cdot [4 + (1-K)(2-K)^2]^2$$

$$P_{K_1, K_2} = K_1 \cdot K_2^6 \cdot (1-K_1)^3 \cdot (2-K_2)^5 \cdot [2 + (1-K_2)(2-K_2)] \cdot [4 + (1-K_2)(2-K_2)^2]^2$$

K constante		K variável	
K	IC95% (K)	K ₁	K ₂
0,564	0,293 - 0,832	-	-

As estimativas de K_1 e K_2 obtidas foram, respectivamente, 0,250 e 1. Como a expressão $(1-K_2)$, referente às gerações III e IV, não ocorre na função, K_2 não é estimado adequadamente. Portanto, o modelo com taxa de penetrância variável não se aplica a essa genealogia.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.3) GENEALOGIA 4

$$P_K = K^6 \cdot (1-K)^4 \cdot (2-K)^3 \cdot [2+(1-K)(2-K)] \cdot [4+(1-K)(2-K)^2]^2 \cdot [8+(1-K)(2-K)^3] \cdot [16+(1-K)(2-K)^4] \cdot [32+(1-K)(2-K)^5]$$

$$P_{K_1, K_2} = K_1^2 \cdot K_2^4 \cdot (1-K_1)^3 \cdot (1-K_2) \cdot (2-K_1) \cdot (2-K_2)^2 \cdot [2+(1-K_1)(2-K_2)] \cdot [4+(1-K_1)(2-K_2)^2] \cdot [4+(1-K_2)(2-K_2)^2] \cdot [8+(1-K_1)(2-K_2)^3] \cdot [16+(1-K_1)(2-K_2)^4] \cdot [32+(1-K_1)(2-K_2)^5]$$

K constante		K variável		G _(1 g.l.)
K	IC95% (K)	K ₁	K ₂	1,028
0,441	0,196 - 0,741	0,313	0,621	(P = 0,311)

Gráfico de log(P_K) e dL/dK

Gráfico de P_K com IC95% (K)

2.4) GENEALOGIA 5

$$P_K = K^2 \cdot (1-K)$$

$$P_{K_1, K_2} = K_2^2 \cdot (1-K_1)$$

K constante		K variável	
K	IC95% (K)	K ₁	K ₂
0,667	0,228 - 0,956	—	—

A genealogia é composta por apenas 3 indivíduos, não possibilitando a aplicação do modelo com K variável.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.5) GENEALOGIA 6

$$P_K = K^2 \cdot (1-K) \cdot (2-K)^9 \cdot [2 + (1-K)(2-K)]$$

$$P_{K_1, K_2} = K_1^2 \cdot (1-K_1) \cdot (2-K_1)^9 \cdot [2 + (1-K_1)(2-K_2)]$$

K constante		K variável	
K	IC95% (K)	K_1	K_2
0,273	0,055 – 0,681	–	–

A estimativa da taxa de penetrância $K_1 = 0.285$ só pode ser obtida se forçarmos o valor de K_2 em zero. Além disso, 12 dos 14 indivíduos da genealogia pertencem à segunda geração,

fornecendo uma estrutura que não é adequada à aplicação do modelo com K variável.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.6) GENEALOGIA 7

$$P_K = K^4 \cdot (1-K)^5 \cdot (2-K)^7 \cdot [4 + (1-K)(2-K)^2]$$

$$P_{K_1, K_2} = K_2^4 \cdot (1-K_1)^4 \cdot (1-K_2) \cdot (2-K_1)^4 \cdot (2-K_2)^3 \cdot [4 + (1-K_2)(2-K_2)^2]$$

K constante		K variável		$G_{(1 \text{ g.l.})}$
K	IC95% (K)	K_1	K_2	7,033
0,322	0,117 - 0,610	0	0,676	(P = 0,008)

Existem oito indivíduos normais nas duas primeiras gerações, sendo quatro deles portadores certos do gene e os restantes, normais sem filhos. Essa estrutura é condizente com a estimativa forçada de $K_1 = 0$, a qual permitiu a estimativa da segunda taxa de penetrância.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.7) GENEALOGIA 8

$$P_K = K^6 \cdot (1-K)^4 \cdot (2-K)^{13}$$

$$P_{K_1, K_2} = K_1^2 \cdot K_2^4 \cdot (1-K_1)^3 \cdot (1-K_2) \cdot (2-K_1) \cdot (2-K_2)^{12}$$

K constante		K variável		G _(1 g.l.)
K	IC95% (K)	K ₁	K ₂	0,039
0,404	0,185 - 0,674	0,371	0,427	(P = 0,843)

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

Nessa genealogia, o fenótipo do ascendente direto dos indivíduos **I-1** e **I-2**, certamente um portador do gene que condiciona a síndrome, não foi descrita.

Para avaliar se a inclusão desse ascendente na genealogia altera significativamente a estimativa de K, refizemos os cálculos considerando duas situações: (A) ascendente penetrante; e (B) ascendente não penetrante.

As funções de verossimilhança e as estimativas de K obtidas para os casos (A) e (B) foram:

$$P_{KA} = K_A^7 \cdot (1-K_A)^4 \cdot (2-K_A)^{13}$$

$$K_A = 0,448$$

$$P_{KB} = K_B^6 \cdot (1-K_B)^5 \cdot (2-K_B)^{13}$$

$$K_B = 0,375$$

Aplicando-se uma propriedade assintótica dos estimadores de máxima verossimilhança, podemos obter um valor médio de K ponderando-se as estimativas K_A e K_B pelos recíprocos dos valores de suas variâncias:

$$\begin{aligned}
K &= (K_A/s_{KA}^2 + K_B/s_{KB}^2) / (1/s_{KA}^2 + 1/s_{KB}^2) \\
&= (K_A \cdot s_{KB}^2 + K_B \cdot s_{KA}^2) / (s_{KA}^2 + s_{KB}^2) \\
&= 0,409 \quad ,
\end{aligned}$$

através, por exemplo, do seguinte código Mathematica:

```

L1 = 7 * Log[K1] + 4 * Log[1-K1] + 13 * Log[2-K1];
L2 = 6 * Log[K2] + 5 * Log[1-K2] + 13 * Log[2-K2];
dL1dK = D[L1,K1]; d2L1dK2 = D[dL1dK,K1];
dL2dK = D[L2,K2]; d2L2dK2 = D[dL2dK,K2];
FindRoot[dL1dK==0,{K1,0.5}]
FindRoot[dL2dK==0,{K2,0.5}]
K1 = 0.448044;
varK1 = -1/d2L1dK2
K2 = 0.375;
varK2 = -1/d2L2dK2
K = (K1/varK1 + K2/varK2)/(1/varK1 + 1/varK2)

{K1 -> 0.448}
{K2 -> 0.375}
{K -> 0.409}

```

Considerando-se que as estimativas de K obtidas (0,404 para a genealogia originalmente descrita e 0,409 para o valor médio de K) não são significativamente diferentes entre si, optamos por usar a genealogia tal como foi apresentada originalmente, sem a especificação do fenótipo do indivíduo gerador da genealogia.

2.8) GENEALOGIA 9

$$P_K = K^4 \cdot (1-K)^5 \cdot (2-K)^4 \cdot [2 + (1-K)(2-K)] \cdot [4 + (1-K)(2-K)^2] \cdot [16 + (1-K)(2-K)^4] \cdot [32 + (1-K)(2-K)^5]$$

$$P_{K_1, K_2} = K_1 \cdot K_2^3 \cdot (1-K_1)^4 \cdot (1-K_2) \cdot (2-K_2)^4 \cdot [2 + (1-K_2)(2-K_2)] \cdot [4 + (1-K_2)(2-K_2)^2] \cdot [16 + (1-K_2)(2-K_2)^4] \cdot [32 + (1-K_2)(2-K_2)^5]$$

K constante		K variável		$G_{(1 \text{ g.l.})}$
K	IC95% (K)	K_1	K_2	0,508
0,303	0,104 - 0,606	0,200	0,410	(P = 0,476)

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.9) GENEALOGIA 10

$$P_K = K^7 \cdot (1-K)^4 \cdot (2-K)^9$$

$$P_{K_1, K_2} = K_1^4 \cdot K_2^3 \cdot (1-K_1)^3 \cdot (1-K_2) \cdot (2-K_1)^2 \cdot (2-K_2)^7$$

K constante		K variável		$G_{(1 \text{ g.l.})}$
K	IC95% (K)	K_1	K_2	0,038
0,500	0,257 - 0,760	0,523	0,466	(P = 0,845)

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.10) GENEALOGIA 11

$$P_K = K^{19} \cdot (1-K)^6 \cdot (2-K)^{23}$$

$$P_{K_1, K_2} = K_1^4 \cdot K_2^{15} \cdot (1-K_1)^6 \cdot (2-K_1)^7 \cdot (2-K_2)^{16}$$

K constante		K variável		$G_{(1 \text{ g.l.})}$
K	IC95% (K)	K_1	K_2	8,485
0,602	0,418 - 0,782	0,311	0,968	(P = 0,004)

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.11) GENEALOGIA 12

$$\begin{aligned}
 P_K &= K^{17} \cdot (1-K)^6 \cdot (2-K)^{31} \cdot [4 + (1-K)(2-K)^2] \cdot [64 + (1-K)(2-K)^6] \\
 &\quad \cdot [18 + 9(1-K)(2-K)^2 + 2(1-K)^2(3-2K)^2] \\
 &\quad \cdot [2304 + 576(1-K)(2-K)^2 + 9(1-K)^2(2-K)^8 + 2(1-K)^3(2-K)^6(3-2K)^2] \\
 P_{K_1, K_2} &= K_1^5 \cdot K_2^{12} \cdot (1-K_1)^5 \cdot (1-K_2) \cdot (2-K_1)^{11} \cdot (2-K_2)^{20} \\
 &\quad \cdot [4 + (1-K_2)(2-K_2)^2] \cdot [64 + (1-K_2)(2-K_2)^6] \\
 &\quad \cdot \{1/4 + (1-K_2)/2 \cdot [(2-K_2)/2]^2 + (1-K_2)/2 \cdot [(3-2K_2)/3]^2\} \\
 &\quad \cdot \{1/4 + 1/2 \cdot (1-K_2)/2 \cdot [(2-K_2)/2]^2 + [(1-K_1)/2] \cdot [(1-K_2)/2] \\
 &\quad \cdot [(2-K_2)/2]^8 + [(1-K_1)/2] \cdot [(1-K_2)/2]^2 \cdot [(2-K_2)/2]^6 \cdot [(3-2K_2)/3]^2\}
 \end{aligned}$$

K constante		K variável		G _(1 g.l.)
K	IC95% (K)	K ₁	K ₂	
0,477	0,304 - 0,672	0,348	0,604	1,610 (P = 0,204)

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

A função de verossimilhança foi obtida analisando-se a genealogia em partes, como detalhado a seguir.

a) árvores sem uniões consanguíneas:

(1)

(2)

(3)

(4)

(5)

(6)

As funções de probabilidades associadas a cada uma dessas árvores são:

$$P_1 = K^2 \cdot (2-K)^9$$

$$P_2 = K^2 \cdot (1-K) \cdot (2-K)^2$$

$$P_3 = K^2 \cdot (1-K)^2 \cdot (2-K)^6$$

$$P_4 = K^2 \cdot (1-K) \cdot (2-K)^9$$

$$P_5 = K^5 \cdot (1-K) \cdot (2-K)$$

$$P_6 = K^4 \cdot (1-K) \cdot (2-K)^4 \cdot [4 + (1-K)(2-K)^2] \cdot [64 + (1-K)(2-K)^6] ,$$

sendo a função de verossimilhança parcial (P_{Ka}) correspondente a esse conjunto de configurações:

$$P_{Ka} = P_1 \cdot P_2 \cdot P_3 \cdot P_4 \cdot P_5 \cdot P_6$$

$$= K^{17} \cdot (1-K)^6 \cdot (2-K)^{31} \cdot [4 + (1-K)(2-K)^2] \cdot [64 + (1-K)(2-K)^6] .$$

b) árvores com uniões consanguíneas:

b.1)

Dado que os indivíduos **IV.9** e **IV.20** são ambos descendentes diretos de heterozigotos certos, as configurações genotípicas possíveis para essa árvore são:

com as seguintes funções de probabilidades:

$$P_1 = 1/2 \cdot 1/2 = 1/4$$

$$P_2 = (1-K)/2 \cdot 1/2 \cdot [(2-K)/2]^2$$

$$P_3 = P_2$$

$$P_4 = [(1-K)/2]^2 \cdot [(3-2K)/3]^2$$

A função de verossimilhança parcial (P_{Kb1}) é dada por

$$\begin{aligned}
 P_{Kb1} &= P_1 + P_2 + P_3 + P_4 \\
 &= 1/4 + (1-K)/2 \cdot \{ [(2-K)/2]^2 + (1-K)/2 \cdot [(3-2K)/3]^2 \} \quad .
 \end{aligned}$$

b.2)

A união consanguínea está inserida em uma árvore de indivíduos normais, cujo ancestral comum (**II.1**) é um heterozigoto certo. Como no caso anterior, identificamos as possíveis configurações genotípicas dessa árvore e suas respectivas funções:

(1)

(2)

(3)

(4)

(5)

$$\begin{aligned}
P_1 &= 1/2 \cdot 1/2 = 1/4 \\
P_2 &= 1/2 \cdot (1-K)/2 \cdot [(2-K)/2]^2 \\
P_3 &= 1/2 \cdot [(1-K)/2]^2 \cdot [(2-K)/2]^8 \\
P_4 &= [(1-K)/2]^3 \cdot [(2-K)/2]^6 \cdot [(3-2K)/3]^2 \\
P_5 &= 1/2 \cdot [(1-K)/2]^2 \cdot [(2-K)/2]^8 = P_3 \quad ,
\end{aligned}$$

cuja função de verossimilhança parcial (P_{Kb2}) é

$$\begin{aligned}
P_{Kb2} &= P_1 + P_2 + P_3 + P_4 + P_5 \\
&= 1/4 + 1/2 \cdot (1-K)/2 \cdot [(2-K)/2]^2 + [(1-K)/2]^2 [(2-K)/2]^8 \\
&\quad + [(1-K)/2]^3 [(2-K)/2]^6 [(3-2K)/3]^2 \quad .
\end{aligned}$$

A função de verossimilhança da genealogia completa é, finalmente, obtida através de:

$$\begin{aligned}
P_K &= P_{Ka} \cdot P_{Kb1} \cdot P_{Kb2} \\
&= K^{17} \cdot (1-K)^6 \cdot (2-K)^{31} \cdot [4 + (1-K)(2-K)^2] \cdot [64 + (1-K)(2-K)^6] \\
&\quad \cdot \{1/4 + (1-K)/2 \cdot \{[(2-K)/2]^2 + (1-K)/2 \cdot [(3-2K)/3]^2\}\} \\
&\quad \cdot \{1/4 + 1/2 \cdot (1-K)/2 \cdot [(2-K)/2]^2 + [(1-K)/2]^2 [(2-K)/2]^8 \\
&\quad + [(1-K)/2]^3 [(2-K)/2]^6 [(3-2K)/3]^2\} \quad ,
\end{aligned}$$

ou, ignorando-se todas as quantidades constantes,

$$\begin{aligned}
P_K &= K^{17} \cdot (1-K)^6 \cdot (2-K)^{31} \cdot [4 + (1-K)(2-K)^2] \cdot [64 + (1-K)(2-K)^6] \\
&\quad \cdot [18 + 9(1-K)(2-K)^2 + 2(1-K)^2(3-2K)^2] \\
&\quad \cdot [2304 + 576(1-K)(2-K)^2 + 9(1-K)^2(2-K)^8 \\
&\quad + 2(1-K)^3(2-K)^6(3-2K)^2] \quad .
\end{aligned}$$

2.12) GENEALOGIA 13

$$P_K = K^2 \cdot (1-K)^2 \cdot (2-K) \cdot [4 + (1-K)(2-K)^2]$$

$$P_{K_1, K_2} = K_1 \cdot K_2 \cdot (1-K_1)^2 \cdot (2-K_1) \cdot [4 + (1-K_1)(2-K_2)^2]$$

K constante		K variável	
K	IC95% (K)	K ₁	K ₂
0,413	0,107 - 0,807	–	–

Conseguimos obter a estimativa **K₁ = 0.280** se forçarmos o valor de **K₂ = 1**, porém essa estimativa só se justificaria se os indivíduos III.1 e III.2 fossem homozigotos normais. Portanto, o modelo com K variável não se aplica a essa genealogia.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.13) GENEALOGIA 14

$$P_K = K^2 \cdot (1-K)^2 \cdot (2-K)^4 \cdot [2 + (1-K)(2-K)] \cdot [4 + (1-K)(2-K)^2]$$

$$P_{K_1, K_2} = K_1 \cdot K_2 \cdot (1-K_1)^2 \cdot (2-K_1)^2 \cdot (2-K_2)^2 \cdot [2 + (1-K_1)(2-K_2)] \cdot [4 + (1-K_1)(2-K_2)]^2$$

K constante		K variável		G _(1 g.l.)
K	IC95% (K)	K ₁	K ₂	0,413 (P = 0,520)
0,294	0,063 – 0,697	0,220	0,503	

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.14) GENEALOGIA 15

$$P_K = K^4 \cdot (1-K)^2 \cdot (2-K)^5 \cdot [512 + (1-K)(2-K)^9] \cdot [4 + (1-K)(2-K)^2] \cdot [8 + (1-K)(2-K)^3]^2 \cdot [128 + (1-K)(2-K)^7]$$

$$P_{K_1, K_2} = K_1^2 \cdot K_2^2 \cdot (1-K_1)^2 \cdot (2-K_1)^3 \cdot (2-K_2)^2 \cdot [512 + (1-K_1)(2-K_2)^9] \cdot [4 + (1-K_1)(2-K_2)^2]^2 \cdot [16 + (1-K_1)(2-K_2)^4] \cdot [128 + (1-K_1)(2-K_2)^7]$$

K constante		K variável		$G_{(1 \text{ g.l.})}$
K	IC95% (K)	K_1	K_2	0,546
0,393	0,132 - 0,777	0,349	0,725	(P = 0,460)

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

A árvore de indivíduos normais **II.8** a **II.11** e **III.14** a **III.28**, tal como representada no trabalho original, não detalha o número de filhos de cada um dos quatro indivíduos da geração II, num total de 15. Existem 592 configurações capazes de explicar essa situação. Em razão disso, deduzimos a função de verossimilhança da genealogia, considerando taxa de penetrância constante entre as gerações, como sendo

$$P_K = K^4 \cdot (1-K)^2 \cdot (2-K)^5 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^9\} \\ \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^i\} \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^j\} \\ \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^l\} \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^m\}$$

e, no caso de taxas variáveis de penetrância (K_1 nas duas primeiras gerações e K_2 na última), como

$$P_{K_1K_2} = K_1^2 \cdot K_2^2 \cdot (1-K_1)^2 \cdot (2-K_1)^3 \cdot (2-K_2)^2 \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^9\} \\ \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^i\} \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^j\} \\ \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^l\} \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^m\} \quad .$$

Nessas funções, i , j , l e m representam os números de filhos dos indivíduos **II.8** a **II.11**, respectivamente, com as restrições: $i + j + l + m = 15$ e $i \leq 9, \dots, m \leq 9$.

Considerando-se uma população de tamanho aproximadamente estável, o tamanho das progênes não deve diferir significativamente de uma distribuição de Poisson com média 2 [$g(x) = e^{-2}2^x/x!$]. Limitamos o número de filhos por casal em nove, pois a partir desse número a probabilidade de Poisson correspondente é praticamente zero.

Valores médios de **K** (taxa de penetrância constante), **K₁** e **K₂** (taxas de penetrância variável) foram obtidos considerando todas as 592 configurações possíveis, a partir das funções:

$$P_K = K^4 \cdot (1-K)^2 \cdot (2-K)^5 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^9\} \\ \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^i\} \cdot e^{-2}2^i/i! \\ \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^j\} \cdot e^{-2}2^j/j! \\ \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^1\} \cdot e^{-2}2^1/1! \\ \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^m\} \cdot e^{-2}2^m/m! \quad e$$

$$P_{K_1, K_2} = K_1^2 \cdot K_2^2 \cdot (1-K_1)^2 \cdot (2-K_1)^3 \cdot (2-K_2)^2 \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^9\} \\ \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^i\} \cdot e^{-2}2^i/i! \\ \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^j\} \cdot e^{-2}2^j/j! \\ \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^1\} \cdot e^{-2}2^1/1! \\ \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^m\} \cdot e^{-2}2^m/m! \quad .$$

Os programas **MAJ15_KC.BAS** e **MAJ15_2K.BAS** (ANEXO I), empregando o método detalhado acima, forneceram as seguintes estimativas:

Modelo		Valor médio	Erro-padrão
K constante		0,3927	0,0046
K variável	K₁	0,3501	0,0018
	K₂	0,7221	0,0475

Dentre todas as combinações possíveis de i, j, l e m, as configurações com **2, 3, 3 e 7** filhos atribuídos aos indivíduos II.8 a II.11 apresentam uma estimativa de K (**0,3935**) que não difere significativamente do valor médio obtido **K = 0,3927**. Por essa razão, tomamos a função de verossimilhança dessa genealogia, sob modelo com K constante, como sendo

$$P_K = K^4 \cdot (1-K)^2 \cdot (2-K)^5 \cdot [512 + (1-K)(2-K)^9] \cdot [4 + (1-K)(2-K)^2] \cdot [8 + (1-K)(2-K)^3]^2 \cdot [128 + (1-K)(2-K)^7] \quad .$$

No modelo com K variável, as configurações que associam **2, 2, 4 e 7** filhos a cada um dos indivíduos II.8 a II.11 apresentam estimativas de **K₁ (0,3486)** e **K₂ (0,7252)**, que não diferem significativamente dos valores médios obtidos **K₁ = 0,3501** e **K₂ = 0,7221**. Assim, sob esse modelo, consideramos a seguinte função de verossimilhança para essa genealogia:

$$P_{K_1 K_2} = K_1^2 \cdot K_2^2 \cdot (1-K_1)^2 \cdot (2-K_1)^3 \cdot (2-K_2)^2 \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^9\} \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^2\}^2 \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^4\} \cdot \{1/2 + (1-K_1)/2 \cdot [(2-K_2)/2]^7\} \quad .$$

2.15) GENEALOGIA 16

$$P_K = K^3 \cdot (1-K)$$

$$P_{K_1, K_2} = K_2^3 \cdot (1-K_1)$$

K constante		K variável	
K	IC95% (K)	K ₁	K ₂
0,750	0,329 - 0,973	–	–

A genealogia é muito pequena, inapropriada ao estudo com K variável.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.16) GENEALOGIA 17

$$P_K = K^2 \cdot (1-K) \cdot (2-K)^4$$

$$P_{K_1, K_2} = K_2^2 \cdot (1-K_1) \cdot (2-K_2)^4$$

K constante		K variável	
K	IC95% (K)	K_1	K_2
0,453	0,124 – 0,866	–	–

A estrutura simples dessa genealogia (um progenitor e seis filhos) não permite, como em alguns casos anteriores, a aplicação do modelo com K variável.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

2.17) GENEALOGIA 18

$$P_K = K^3 \cdot (1-K)^5 \cdot (2-K)^7 \cdot [16 + (1-K)(2-K)^4]$$

$$\cdot [1458(2-K)^4 + 5832(2-K)^2 + 64(1-K)^2(3-2K)^5 + 729(1-K)(2-K)^5]$$

$$P_{K_1, K_2} = K_1 \cdot K_2 \cdot (1-K_1)^2 \cdot (1-K_2) \cdot (2-K_2)^3 \cdot [16 + (1-K_1)(2-K_2)^4]$$

$$\cdot \{1/2 \cdot K_2/2 \cdot (1-K_1)/2 \cdot (1-K_2)/2 \cdot (2-K_2)^8/256$$

$$+ 1/2 \cdot K_2/2 \cdot (1-K_1)/2 \cdot (1-K_2)/2 \cdot (2-K_2)^6/64$$

$$+ 2K_2/3 \cdot (1-K_1)^2/4 \cdot (1-K_2)^2/4 \cdot (2-K_2)^4/16 \cdot (3-2K_2)^5/243$$

$$+ K_2/2 \cdot (1-K_1)^2/4 \cdot (1-K_2)/2 \cdot (2-K_2)^9/512\}$$

K constante		K variável		G _(1 g.l.)
K	IC95% (K)	K ₁	K ₂	
0,214	0,057 - 0,493	0,215	0,213	0,00013 (P = 0,991)

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

Para construir a função de verossimilhança, a genealogia foi analisada em partes.

a) **árvore sem união consanguínea (incluindo o indivíduo II-3, ancestral comum da árvore com união consanguínea):**

cuja função de verossimilhança parcial (P_{Ka}) é

$$P_{Ka} = K^2 \cdot (1-K)^3 \cdot (2-K)^3 \cdot [16 + (1-K)(2-K)^4] \quad .$$

b) **árvore com união consanguínea:**

As configurações genotípicas capazes de explicar essa árvore e suas respectivas funções de probabilidades são:

(1)

(2)

(3)

(4)

(5)

$$P_1 = 1/2 \cdot K/2 \cdot (1-K)^2/4 \cdot (2-K)^8/256$$

$$P_2 = 1/2 \cdot K/2 \cdot (1-K)^2/4 \cdot (2-K)^6/64$$

$$P_3 = 2K/3 \cdot (1-K)^4/16 \cdot (2-K)^4/16 \cdot [(3-2K)/3]^5$$

$$P_4 = P_5 = 1/2 \cdot K/2 \cdot (1-K)^3/8 \cdot (2-K)^9/512$$

A função de verossimilhança parcial (P_{Kb}) é, portanto, dada por

$$P_{Kb} = P_1 + P_2 + P_3 + P_4 + P_5$$

$$\begin{aligned} P_{Kb} = & 1/2 \cdot K/2 \cdot (1-K)^2/4 \cdot (2-K)^8/256 + 1/2 \cdot K/2 \cdot (1-K)^2/4 \cdot (2-K)^6/64 \\ & + 2K/3 \cdot (1-K)^4/16 \cdot (2-K)^4/16 \cdot [(3-2K)/3]^5 \\ & + K/2 \cdot (1-K)^3/8 \cdot (2-K)^9/512 \end{aligned}$$

Finalmente, a função de verossimilhança dessa família é

$$P_K = P_{Ka} \cdot P_{Kb}$$

$$P_K = K^2 \cdot (1-K)^3 \cdot (2-K)^3 \cdot [16 + (1-K)(2-K)^4] \\ \cdot \{1/2 \cdot K/2 \cdot (1-K)^2/4 \cdot (2-K)^8/256 + 1/2 \cdot K/2 \cdot (1-K)^2/4 \cdot (2-K)^6/64 \\ + 2K/3 \cdot (1-K)^4/16 \cdot (2-K)^4/16 \cdot [(3-2K)/3]^5 \\ + K/2 \cdot (1-K)^3/8 \cdot (2-K)^9/512\} ,$$

ou, ignorando-se todas as quantidades constantes,

$$P_K = K^3 \cdot (1-K)^5 \cdot (2-K)^7 \cdot [16 + (1-K)(2-K)^4] \\ \cdot [1458(2-K)^4 + 5832(2-K)^2 + 64(1-K)^2(3-2K)^5 + 729(1-K)(2-K)^5] .$$

2.18) GENEALOGIA 19

$$P_K = K^{15} \cdot (1-K)^{26} \cdot (2-K)^7 \cdot [4 + (1-K)(2-K)^2] \cdot [32 + (1-K)(2-K)^5] \\ \cdot [6 + 3(1-K) + 3(1-K)^2 + 2(1-K)^3] \cdot \{243 \cdot \{512 + (1-K) \cdot (2-K) \\ \cdot \{2 + (1-K)(2-K) \cdot [4 + (1-K)(2-K)^2]^2\} \cdot [8 + (1-K)(2-K)^3] \\ \cdot \{16 + (1-K) \cdot [8 + (1-K)(2-K)^3]\}\}$$

$$\begin{aligned}
& + 512 \cdot (1-K) \cdot \{256 + (1-K)(2-K) \cdot [2 + (1-K) \cdot (2-K)] \cdot [4 + (1-K)(2-K)^2]^2\} \\
& \cdot [4 + (1-K)(2-K)^3] \cdot \{8 + (1-K)[8 + (1-K)(2-K)^3]\} \} \\
P_{K_1, K_2} = & K_2^9 \cdot (1-K_1)^{17} \cdot (1-K_2)^2 \cdot (2-K_2)^3 \cdot \{1/2 \cdot K_2/2 \cdot [(1-K_1)/2]^3 \cdot [(1-K_2)/2]^3 \\
& \cdot (2-K_2)/2 + 1/2 \cdot K_2/2 \cdot [(1-K_1)/2] \cdot [(1-K_2)/2]^3 \cdot (2-K_2)/2 + 1/2 \cdot K_2/2 \\
& \cdot [(1-K_1)/2]^2 \cdot [(1-K_2)/2]^3 \cdot (2-K_2)/2 + 1/2 \cdot K_2/2 \cdot [(1-K_1)/2]^3 \\
& \cdot [(1-K_2)/2]^3 \cdot (2-K_2)/2 + K_2/2 \cdot [(1-K_1)/2]^4 \cdot 2(1-K_2)/3 \cdot [(1-K_2)/2]^2 \\
& \cdot (2-K_2)/2\} \cdot \{(1-K_1)/2 \cdot \{1/2 + (1-K_2)/2 \cdot [1/2 + (1-K_2)/2 \cdot (2-K_2)/2] \\
& \cdot \{1/2 + (1-K_2)/2 \cdot [(2-K_2)/2]^2\}^2 \cdot (2-K_2)/2\} \cdot \{1/2 + (1-K_2)/2 \\
& \cdot [(2-K_2)/2]^3\} \cdot (1-K_2)/2 \cdot (K_2/2)^2 \cdot (2-K_2)/2 \cdot \{(1-K_2)/2 \\
& \cdot \{1/2 + (1-K_2)/2 \cdot [(2-K_2)/2]^2\} \cdot (K_2/2)^3 \cdot (2-K_2)/2\}^2 \\
& \cdot \{1/2 + (1-K_2)/2 \cdot [(2-K_2)/2]^5\} \cdot \{1/2 + (1-K_2)/2 \cdot \{1/2 + (1-K_2)/2 \\
& \cdot [(2-K_2)/2]^3\} + [(1-K_1)/2]^2 \cdot \{1/3 + 2(1-K_2)/3 \cdot [1/2 + (1-K_2)/2 \\
& \cdot (2-K_2)/2] \cdot \{1/2 + (1-K_2)/2 \cdot [(2-K_2)/2]^2\}^2 \cdot (2-K_2)/2\} \\
& \cdot \{1/3 + 2(1-K_2)/3 \cdot [(2-K_2)/2]^3\} \cdot 2(1-K_2)/3 \cdot (K_2/2)^2 \cdot (2-K_2)/2 \\
& \cdot \{2(1-K_2)/3 \cdot \{1/2 + (1-K_2)/2 \cdot [(2-K_2)/2]^2\} \cdot (K_2/2)^3 \cdot [(2-K_2)/2]^2 \\
& \cdot \{1/2 + (1-K_2)/2 \cdot [(2-K_2)/2]^5\} \cdot \{1/3 + 2(1-K_2)/3 \cdot \{1/2 + (1-K_2)/2 \\
& \cdot [(2-K_2)/2]^3\} \} \} \}
\end{aligned}$$

K constante		K variável	
K	IC95% (K)	K ₁	K ₂
0,311	0,183 - 0,443	0	0,484

Essa genealogia é composta por 80 indivíduos distribuídos em 9 gerações. As estimativas de **K₁** e **K₂** foram obtidas considerando-se as 5 primeiras gerações com **K₁** e as restantes, com **K₂**. Nesse caso, a estimativa **K₁ = 0**, apesar de forçada (no sentido de ter sido tratada como constante para poder obter-se a estimativa de **K₂**) é perfeitamente aceitável, uma vez que nas 5 primeiras gerações encontram-se apenas indivíduos normais, a maioria dos quais portadores certos do gene mutado.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

A genealogia foi dividida nas seguintes partes para elaboração da função de verossimilhança:

a) árvores sem uniões consanguíneas:

As funções de probabilidades associadas a essas configurações são, respectivamente:

$$P_1 = (1-K)$$

$$P_2 = K \cdot (1-K)^5 \cdot (2-K)$$

$$P_3 = K^2 \cdot (1-K)^4$$

$$P_4 = K^2 \cdot (1-K)^5 \cdot (2-K)$$

$$P_5 = K^4 \cdot (1-K)^4 \cdot (2-K) ,$$

sendo a função de verossimilhança parcial (P_{Ka})

$$P_{Ka} = P_1 \cdot P_2 \cdot P_3 \cdot P_4 \cdot P_5$$

$$= K^9 \cdot (1-K)^{19} \cdot (2-K)^3 .$$

b) árvores com uniões consanguíneas:

b.1)

As possíveis configurações genotípicas capazes de explicar essa árvore são:

com as respectivas funções de probabilidades:

$$P_1 = 1/2 \cdot K/2 \cdot [(1-K)/2]^6 \cdot (2-K)/2$$

$$P_2 = 1/2 \cdot K/2 \cdot [(1-K)/2]^4 \cdot (2-K)/2$$

$$P_3 = 1/2 \cdot K/2 \cdot [(1-K)/2]^5 \cdot (2-K)/2$$

$$P_4 = 1/2 \cdot K/2 \cdot [(1-K)/2]^6 \cdot (2-K)/2$$

$$P_5 = K/2 \cdot [(1-K)/2]^6 \cdot 2(1-K)/3 \cdot (2-K)/2 \quad .$$

A função de verossimilhança parcial (P_{Kb1}) é dada por

$$P_{Kb1} = P_1 + P_2 + P_3 + P_4 + P_5$$

$$= K/2 \cdot [(1-K)/2]^4 \cdot (2-K)/2 \cdot \{1/2 \cdot [(1-K)/2]^2 + 1/2$$

$$+ 1/2 \cdot (1-K)/2 + 1/2 \cdot [(1-K)/2]^2 + [(1-K)/2]^2 \cdot 2(1-K)/3\} \quad .$$

b.2)

As possíveis configurações e as respectivas funções de probabilidades associadas a essa árvore são:

(1)

$$\begin{aligned}
 P_1 = & 1/2 \cdot (1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [1/2 + (1-K)/2 \cdot (2-K)/2] \\
 & \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\}^2 \cdot (2-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\} \\
 & \cdot (1-K)/2 \cdot (K/2)^2 \cdot (2-K)/2 \\
 & \cdot \{(1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\} \cdot (K/2)^3 \cdot [(2-K)/2]^2 \\
 & \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^5\}\} \\
 & \cdot \{1/2 + (1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\}\} \quad .
 \end{aligned}$$

A segunda configuração possível é obtida alterando-se a condição de heterozigose do indivíduo **IV-4** para o indivíduo **V-5**. A função de probabilidades é igual à estabelecida acima, ou seja, $P_2 = P_1$.

(3)

$$\begin{aligned}
 P_3 = & [(1-K)/2]^2 \cdot \{1/3 + 2(1-K)/3 \cdot [1/2 + (1-K)/2 \cdot (2-K)/2] \\
 & \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\}^2 \cdot (2-K)/2\} \\
 & \cdot \{1/3 + 2(1-K)/3 \cdot [(2-K)/2]^3\} \cdot 2(1-K)/3 \cdot (K/2)^2 \cdot (2-K)/2 \\
 & \cdot \{2(1-K)/3 \cdot [1/2 + (1-K)/2 \cdot [(2-K)/2]^2] \cdot (K/2)^3 \cdot [(2-K)/2]^2 \\
 & \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^5\} \\
 & \cdot \{1/3 + 2(1-K)/3 \cdot [1/2 + (1-K)/2 \cdot [(2-K)/2]^3]\} \quad ,
 \end{aligned}$$

sendo a função de verossimilhança parcial (P_{Kb2}):

$$\begin{aligned}
 P_{Kb2} = & P_1 + P_2 + P_3 \\
 = & (1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [1/2 + (1-K)/2 \cdot (2-K)/2] \\
 & \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\}^2 \cdot (2-K)/2\} \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\} \\
 & \cdot (1-K)/2 \cdot (K/2)^2 \cdot (2-K)/2 \cdot \{(1-K)/2 \cdot [1/2 + (1-K)/2 \\
 & \cdot [(2-K)/2]^2] \cdot (K/2)^3 \cdot (2-K)/2\}^2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^5\} \\
 & \cdot \{1/2 + (1-K)/2 \cdot [1/2 + (1-K)/2 \cdot [(2-K)/2]^3]\} \\
 + & [(1-K)/2]^2 \cdot \{1/3 + 2(1-K)/3 \cdot [1/2 + (1-K)/2 \cdot (2-K)/2] \\
 & \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\}^2 \cdot (2-K)/2\} \\
 & \cdot \{1/3 + 2(1-K)/3 \cdot [(2-K)/2]^3\} \cdot 2(1-K)/3 \cdot (K/2)^2 \cdot (2-K)/2 \\
 & \cdot \{2(1-K)/3 \cdot [1/2 + (1-K)/2 \cdot [(2-K)/2]^2] \cdot (K/2)^3 \cdot [(2-K)/2]^2
 \end{aligned}$$

$$\begin{aligned}
& \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^5\} \\
& \cdot \{1/3 + 2(1-K)/3 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\}\} \quad .
\end{aligned}$$

Enfim, a função de verossimilhança da família é obtida através de:

$$\begin{aligned}
P_K &= P_{Ka} \cdot P_{Kb1} \cdot P_{Kb2} \\
&= K^9 \cdot (1-K)^{19} \cdot (2-K)^3 \\
&\cdot \{K/2 \cdot [(1-K)/2]^4 \cdot (2-K)/2 \cdot \{1/2 \cdot [(1-K)/2]^2 + 1/2 \\
&+ 1/2 \cdot (1-K)/2 + 1/2 \cdot [(1-K)/2]^2 + [(1-K)/2]^2 \cdot 2(1-K)/3\} \\
&\cdot \{(1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [1/2 + (1-K)/2 \cdot (2-K)/2] \\
&\cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\}^2 \cdot (2-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\} \\
&\cdot (1-K)/2 \cdot (K/2)^2 \cdot (2-K)/2 \\
&\cdot \{(1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\} \cdot (K/2)^3 \cdot (2-K)/2\}^2 \\
&\cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^5\} \\
&\cdot \{1/2 + (1-K)/2 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\}\} \\
&+ [(1-K)/2]^2 \cdot \{1/3 + 2(1-K)/3 \cdot [1/2 + (1-K)/2 \cdot (2-K)/2] \\
&\cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\}^2 \cdot (2-K)/2\} \\
&\cdot \{1/3 + 2(1-K)/3 \cdot [(2-K)/2]^3\} \cdot 2(1-K)/3 \cdot (K/2)^2 \cdot (2-K)/2 \\
&\cdot \{2(1-K)/3 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\} \cdot (K/2)^3 \cdot [(2-K)/2]^2 \\
&\cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^5\} \\
&\cdot \{1/3 + 2(1-K)/3 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\}\} \quad ,
\end{aligned}$$

ou, ignorando-se todas as quantidades constantes,

$$\begin{aligned}
P_K &= K^{15} \cdot (1-K)^{26} \cdot (2-K)^7 \cdot [4 + (1-K)(2-K)^2] \cdot [32 + (1-K)(2-K)^5] \\
&\cdot [6 + 3(1-K) + 3(1-K)^2 + 2(1-K)^3] \cdot \{243 \cdot \{512 + (1-K)(2-K) \\
&\cdot \{2 + (1-K)(2-K) \cdot [4 + (1-K)(2-K)^2]^2\} \cdot [8 + (1-K)(2-K)^3] \\
&\cdot \{16 + (1-K)[8 + (1-K)(2-K)^3]\}\} \\
&+ 512(1-K) \cdot \{256 + (1-K)(2-K) \cdot [2 + (1-K)(2-K)] \cdot [4 + (1-K)(2-K)^2]^2\} \\
&\cdot [4 + (1-K)(2-K)^3] \cdot \{8 + (1-K) \cdot [8 + (1-K)(2-K)^3]\} \quad .
\end{aligned}$$

2.19) GENEALOGIA 20

$$\begin{aligned}
 P_K &= K^{23} \cdot (1-K)^{26} \cdot (2-K)^{31} \cdot [8 + (1-K)(2-K)^3] \cdot [64 + (1-K)(2-K)^6] \\
 &\quad \cdot \{6561 \cdot \{2-K\}^5 \cdot [2 + (1-K) + (1-K)^2] + 1024(3-2K)^5(1-K)^2 \cdot [1 + (1-K)]\} \\
 &\quad + \{(2-K)^2 \cdot [729 + 243(1-K)] + (1-K)(3-2K) \cdot [324(1-K)(2-K) \\
 &\quad + 216(2-K) + 96(1-K)^2(3-2K)]\}
 \end{aligned}$$

$$\begin{aligned}
 P_{K_1, K_2} &= K_1^3 \cdot K_2^{15} \cdot (1-K_1)^{16} \cdot (1-K_2)^5 \cdot (2-K_1)^2 \cdot (2-K_2)^{25} \cdot [8 + (1-K_2)(2-K_2)^3] \\
 &\quad \cdot [64 + (1-K_1) \cdot (2-K_2)^6] \\
 &\quad \cdot \{1/4 \cdot (K_2/2)^3 \cdot [(1-K_1)/2] \cdot [(1-K_2)/2]^2 \cdot [(2-K_2)/2]^9 \\
 &\quad + 1/2 \cdot (K_2/2)^3 \cdot [(1-K_1)/2] \cdot [(1-K_2)/2]^3 \cdot [(2-K_2)/2]^9 \\
 &\quad + 1/2 \cdot (K_2/2)^3 \cdot [(1-K_1)/2] \cdot [(1-K_2)/2] \cdot [(2-K_2)/2]^9 \\
 &\quad + 1/4 \cdot (K_2/2)^3 \cdot [(1-K_1)/2]^2 \cdot [(1-K_2)/2] \cdot [(2-K_2)/2]^9 \\
 &\quad + 1/2 \cdot K_2/2 \cdot (2K_2/3)^2 \cdot [(1-K_1)/2]^2 \cdot [(1-K_2)/2] \cdot 2(1-K_2)/3 \\
 &\quad \cdot [(2-K_2)/2]^4 \cdot [(3-2K_2)/3]^5 \\
 &\quad + 1/2 \cdot (K_2/2)^3 \cdot [(1-K_1)/2]^2 \cdot [(1-K_2)/2]^2 \cdot [(2-K_2)/2]^9 \\
 &\quad + K_2/2 \cdot (2K_2/3)^2 \cdot [(1-K_1)/2]^2 \cdot [(1-K_2)/2]^2 \cdot 2(1-K_2)/3 \\
 &\quad \cdot [(2-K_2)/2]^4 \cdot [(3-2K_2)/3]^5\} \\
 &\quad \cdot \{1/2 \cdot (K_2/2)^2 \cdot [(1-K_2)/2]^3 \cdot [(2-K_2)/2]^2 \\
 &\quad + 1/2 \cdot (K_2/2)^2 \cdot [(1-K_2)/2]^4 \cdot [(2-K_2)/2]^2 \\
 &\quad + 3/2 \cdot K_2/2 \cdot 2K_2/3 \cdot [(1-K_2)/2]^5 \cdot (3-2K_2)/3 \cdot (2-K_2)/2 \\
 &\quad + 1/4 \cdot (K_2/2)^2 \cdot [(1-K_2)/2]^3 \cdot [(2-K_2)/2]^2
 \end{aligned}$$

$$+ 1/2 \cdot K_2/2 \cdot 2K_2/3 \cdot [(1-K_2)/2]^4 \cdot (3-2K_2)/3 \cdot (2-K_2)/2$$

$$+ [(1-K_2)/2]^6 \cdot [2K_2/3 \cdot (3-2K_2)/3]^2\}$$

K constante		K variável		G _(1 g.l.)
K	IC95% (K)	K ₁	K ₂	21,861
0,344	0,234 – 0,470	0,140	0,460	(P < 0,001)

Gráfico de log(P_K) e dL/dK

Gráfico de P_K com IC95% (K)

Como no caso anterior, a genealogia foi analisada em várias partes para elaboração da função de verossimilhança.

a) árvores sem uniões consanguíneas:

(1)

(2)

(3)

(4)

(5)

As funções de probabilidades associadas a essas configurações são, respectivamente:

$$P_1 = K^{11} \cdot (1-K)^{12} \cdot (2-K)^{11} \cdot [8 + (1-K)(2-K)^3]$$

$$P_2 = [64 + (1-K) \cdot (2-K)^6]$$

$$P_3 = K^2 \cdot (1-K)^2 \cdot (2-K)^5$$

$$P_4 = K \cdot (1-K)^3 \cdot (2-K)^{10}$$

$$P_5 = K^4 \cdot (1-K)^4 \cdot (2-K) \quad ,$$

cuja função de verossimilhança parcial (P_{Ka}) é

$$\begin{aligned} P_{Ka} &= P_1 \cdot P_2 \cdot P_3 \cdot P_4 \cdot P_5 \cdot P_6 \\ &= K^{18} \cdot (1-K)^{21} \cdot (2-K)^{27} \cdot [8 + (1-K)(2-K)^3] \cdot [64 + (1-K)(2-K)^6] \quad . \end{aligned}$$

b) árvores com uniões consanguíneas:

A genealogia possui três árvores derivadas de uniões consanguíneas, duas das quais estão interligadas:

b.1)

Essa estrutura pode ser explicada através das seguintes configurações:

(1)

(2)

(3)

(4)

(5)

(6)

(7) ,

cujas funções de probabilidades são, respectivamente:

$$P_1 = P_4 = 1/4 \cdot (K/2)^3 \cdot [(1-K)/2]^3 \cdot [(2-K)/2]^9$$

$$P_2 = P_6 = 1/2 \cdot (K/2)^3 \cdot [(1-K)/2]^4 \cdot [(2-K)/2]^9$$

$$P_3 = 1/2 \cdot (K/2)^3 \cdot [(1-K)/2]^2 \cdot [(2-K)/2]^9$$

$$P_5 = 1/2 \cdot K/2 \cdot (2K/3)^2 \cdot [(1-K)/2]^3 \cdot 2(1-K)/3 \cdot [(2-K)/2]^4 \cdot [(3-2K)/3]^5$$

$$P_7 = K/2 \cdot (2K/3)^2 \cdot [(1-K)/2]^4 \cdot 2(1-K)/3 \cdot [(2-K)/2]^4 \cdot [(3-2K)/3]^5$$

A função de verossimilhança parcial (P_{kb1}) é, portanto, dada por

$$\begin{aligned} P_{kb1} &= P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 \\ &= K/2 \cdot [(1-K)/2]^2 \cdot [(2-K)/2]^4 \cdot \{1/2 \cdot (K/2)^2 \cdot (1-K)/2 \cdot [(2-K)/2]^5 \\ &\quad + (K/2)^2 \cdot [(1-K)/2]^2 \cdot [(2-K)/2]^5 + 1/2 \cdot (K/2)^2 \cdot [(2-K)/2]^5 \\ &\quad + 1/2 \cdot (2K/3)^2 \cdot (1-K)/2 \cdot 2(1-K)/3 \cdot [(3-2K)/3]^5 \\ &\quad + (2K/3)^2 \cdot [(1-K)/2]^2 \cdot 2(1-K)/3 \cdot [(3-2K)/3]^5\} \end{aligned}$$

b.2)

As possíveis configurações genotípicas dessa árvore e suas respectivas funções de probabilidades são:

(1)

(2)

(3)

(4)

(5)

(6)

(7)

(8)

(9)

$$\begin{aligned}
P_1 &= 1/2 \cdot (K/2)^2 \cdot [(1-K)/2]^3 \cdot [(2-K)/2]^2 \\
P_2 &= P_7 = 1/4 \cdot (K/2)^2 \cdot [(1-K)/2]^4 \cdot [(2-K)/2]^2 \\
P_3 &= P_4 = P_8 = 1/2 \cdot K/2 \cdot 2K/3 \cdot [(1-K)/2]^5 \cdot (3-2K)/3 \cdot (2-K)/2 \\
P_5 &= 1/4 \cdot (K/2)^2 \cdot [(1-K)/2]^3 \cdot [(2-K)/2]^2 \\
P_6 &= 1/2 \cdot K/2 \cdot 2K/3 \cdot [(1-K)/2]^4 \cdot (3-2K)/3 \cdot (2-K)/2 \\
P_9 &= [(1-K)/2]^6 \cdot [2K/3 \cdot (3-2K)/3]^2 ,
\end{aligned}$$

cuja função de verossimilhança parcial (P_{Kb2}) é dada por

$$\begin{aligned}
P_{Kb2} &= P_1 + P_2 + P_3 + P_4 + P_5 + P_6 + P_7 + P_8 + P_9 \\
P_{Kb2} &= 1/2 \cdot (K/2)^2 \cdot [(1-K)/2]^3 \cdot [(2-K)/2]^2 \\
&\quad + 1/2 \cdot (K/2)^2 \cdot [(1-K)/2]^4 \cdot [(2-K)/2]^2 \\
&\quad + 3/2 \cdot K/2 \cdot 2K/3 \cdot [(1-K)/2]^5 \cdot (3-2K)/3 \cdot (2-K)/2 \\
&\quad + 1/4 \cdot (K/2)^2 \cdot [(1-K)/2]^3 \cdot [(2-K)/2]^2 \\
&\quad + 1/2 \cdot K/2 \cdot 2K/3 \cdot [(1-K)/2]^4 \cdot (3-2K)/3 \cdot (2-K)/2 \\
&\quad + [(1-K)/2]^6 \cdot [2K/3 \cdot (3-2K)/3]^2 .
\end{aligned}$$

A função de verossimilhança da família é, portanto, obtida através de

$$\begin{aligned}
P_K &= P_{Ka} \cdot P_{Kb1} \cdot P_{Kb2} \\
P_K &= K^{18} \cdot (1-K)^{21} \cdot (2-K)^{27} \cdot [8 + (1-K)(2-K)^3] \cdot [64 + (1-K)(2-K)^6] \\
&\quad \cdot \{K/2 \cdot [(1-K)/2]^2 \cdot [(2-K)/2]^4 \cdot \{1/2 \cdot (K/2)^2 \cdot (1-K)/2 \cdot [(2-K)/2]^5 \\
&\quad + (K/2)^2 \cdot [(1-K)/2]^2 \cdot [(2-K)/2]^5 + 1/2 \cdot (K/2)^2 \cdot [(2-K)/2]^5 \\
&\quad + 1/2 \cdot (2K/3)^2 \cdot (1-K)/2 \cdot 2(1-K)/3 \cdot [(3-2K)/3]^5 \\
&\quad + (2K/3)^2 \cdot [(1-K)/2]^2 \cdot 2(1-K)/3 \cdot [(3-2K)/3]^5\} \} \\
&\quad \cdot \{1/2 \cdot (K/2)^2 \cdot [(1-K)/2]^3 \cdot [(2-K)/2]^2 \\
&\quad + 1/2 \cdot (K/2)^2 \cdot [(1-K)/2]^4 \cdot [(2-K)/2]^2 \\
&\quad + 3/2 \cdot K/2 \cdot 2K/3 \cdot [(1-K)/2]^5 \cdot (3-2K)/3 \cdot (2-K)/2 \\
&\quad + 1/4 \cdot (K/2)^2 \cdot [(1-K)/2]^3 \cdot [(2-K)/2]^2 \\
&\quad + 1/2 \cdot K/2 \cdot 2K/3 \cdot [(1-K)/2]^4 \cdot (3-2K)/3 \cdot (2-K)/2 \\
&\quad + [(1-K)/2]^6 \cdot [2K/3 \cdot (3-2K)/3]^2\} ,
\end{aligned}$$

ou, ignorando-se todas as quantidades constantes,

$$P_K = K^{23} \cdot (1-K)^{26} \cdot (2-K)^{31} \cdot [8 + (1-K)(2-K)^3] \cdot [64 + (1-K)(2-K)^6] \\ \cdot \{6561 \cdot (2-K)^5 \cdot [2 + (1-K) + (1-K)^2] + 1024(3-2K)^5(1-K)^2 \cdot [1 + (1-K)]\} \\ + \{(2-K)^2 \cdot [729 + 243(1-K)] + (1-K)(3-2K) \cdot [324(1-K)(2-K) \\ + 216(2-K) + 96(1-K)^2(3-2K)]\} \quad .$$

2.20) GENEALOGIA 21

$$P_K = K^3 \cdot (1-K)^3 \cdot (2-K)^3 \cdot (3-3K+K^2) \cdot [4 + (1-K)(2-K)^2] \\ \cdot [32 + (1-K)(2-K)^5]^2 \cdot [64 + (1-K)(2-K)^6]^3 \cdot [512 + (1-K)(2-K)^9] \\ P_{K_2} = K_2^3 \cdot (2-K_2)^3 \cdot (3-3K_2+K_2^2) \cdot [4 + (2-K_2)^2] \cdot [32 + (2-K_2)^5]^3 \\ \cdot [64 + (2-K_2)^6] \cdot [256 + (2-K_2)^8]^2 \quad .$$

K constante		K variável		G _(1 g.l.)
K	IC95% (K)	K ₁	K ₂	
0,191	0,044 - 0,574	0	0,354	2,937 (P = 0,087)

A genealogia é composta por 57 indivíduos distribuídos em 3 gerações. Nas 2 primeiras gerações, que foram utilizadas para obter a estimativa de **K₁**, existem 10 indivíduos normais: 3 heterozigotos certos e 7 progenitores de árvores normais. Essa

estrutura é compatível com a estimativa $K_1 = 0$ (forçada). Portanto, essa família foi mantida para o estudo com K variável.

Gráfico de $\log(P_K)$ e dL/dK

Gráfico de P_K com IC95% (K)

As árvores de indivíduos normais **II.2** a **II.8** e **III.5** a **III.43** foram representadas no trabalho original de forma agrupada sem detalhar o número de filhos de cada um dos sete indivíduos da geração II, num total de 39 mencionados. Para lidar com essa situação, utilizamos o mesmo procedimento aplicado à genealogia 15 [família 4 descrita por MAJEWSKI et al. (1985)].

A função de verossimilhança dessa genealogia, admitindo-se uma taxa de penetrância constante entre as gerações, é definida como:

$$P_K = K^3 \cdot (1-K)^3 \cdot (2-K)^3 \cdot (3-3K+K^2) \\ \cdot \{1/2+(1-K)/2 \cdot [(2-K)/2]^i\} \cdot \{1/2+(1-K)/2 \cdot [(2-K)/2]^j\} \\ \cdot \{1/2+(1-K)/2 \cdot [(2-K)/2]^l\} \cdot \{1/2+(1-K)/2 \cdot [(2-K)/2]^m\} \\ \cdot \{1/2+(1-K)/2 \cdot [(2-K)/2]^n\} \cdot \{1/2+(1-K)/2 \cdot [(2-K)/2]^o\} \\ \cdot \{1/2+(1-K)/2 \cdot [(2-K)/2]^p\}$$

e, considerando-se taxas de penetrância variáveis de acordo com as gerações (duas primeiras gerações com K_1 e a última, com K_2), como:

$$P_{K_1, K_2} = K_2^3 \cdot (1-K_1)^3 \cdot (2-K_2)^3 \cdot (3-3K_2+K_2^2) \\ \cdot \{1/2+(1-K_1)/2 \cdot [(2-K_2)/2]^i\} \cdot \{1/2+(1-K_1)/2 \cdot [(2-K_2)/2]^j\} \\ \cdot \{1/2+(1-K_1)/2 \cdot [(2-K_2)/2]^l\} \cdot \{1/2+(1-K_1)/2 \cdot [(2-K_2)/2]^m\} \\ \cdot \{1/2+(1-K_1)/2 \cdot [(2-K_2)/2]^n\} \cdot \{1/2+(1-K_1)/2 \cdot [(2-K_2)/2]^o\} \\ \cdot \{1/2+(1-K_1)/2 \cdot [(2-K_2)/2]^p\}$$

Nas funções acima, i , j , l , m , n , o e p são os números de filhos de cada um dos sete indivíduos normais da geração II, com as restrições: $i + j + l + m + n + o + p = 39$ e $i \leq 9, \dots, p \leq 9$ (com esse número, a probabilidade de Poisson correspondente é zero para uma população de tamanho estável com um número médio de filhos da ordem de 2 a 3). Aplicando-se essas restrições, o número de configurações genealógicas compatíveis com as restrições acima é de 326.869.

Utilizando o mesmo raciocínio aplicado na análise da genealogia 15, empregamos os programas **MAJ21_KC.BAS** e **MAJ21_2K.BAS (ANEXO I)** para obter as seguintes estimativas:

Modelo		Valor médio	Erro-padrão
K constante		0,1908	0,0019
K variável	K ₁	–	–
	K ₂	0,3536	0,0169

Entre todas as possíveis combinações para i, j, l, m, n, o e p, as configurações que apresentam as combinações de **2, 5, 5, 6, 6, 6** e **9** filhos para cada um dos sete indivíduos da geração II fornecem uma estimativa de **K (0,19082)** que não difere significativamente do valor médio **0,1908**. Portanto, definimos a função de verossimilhança dessa genealogia, sob modelo com K constante, como sendo:

$$P_K = K^3 \cdot (1-K)^3 \cdot (2-K)^3 \cdot (3-3K+K^2) \cdot [4+(1-K)(2-K)^2] \cdot [32+(1-K)(2-K)^5]^2 \cdot [64+(1-K)(2-K)^6]^3 \cdot [512+(1-K)(2-K)^9]$$

Considerando-se o modelo com K variável, as configurações com combinações de **2, 5, 5, 5, 6, 8** e **8** filhos para os indivíduos II.2 a II.8 apresentam uma estimativa de **K₂ (0,35388)** que não difere significativamente do valor médio **0,3536** obtido, forçando-se a estimativa **K₁ = 0**. Por essa razão, a função de verossimilhança é definida como sendo:

$$\begin{aligned} P_{K_1, K_2} &= K_2^3 \cdot (1-K_1)^3 \cdot (2-K_2)^3 \cdot (3-3K_2+K_2^2) \cdot [4+(1-K_1)(2-K_2)^2] \\ &\cdot [32+(1-K_1)(2-K_2)^5]^3 \cdot [64+(1-K_1)(2-K_2)^6] \cdot [256+(1-K_1)(2-K_2)^8]^2 \\ &= P_{K_2} \\ &= K_2^3 \cdot (2-K_2)^3 \cdot (3-3K_2+K_2^2) \cdot [4+(2-K_2)^2] \cdot [32+(2-K_2)^5]^3 \\ &\cdot [64+(2-K_2)^6] \cdot [256+(2-K_2)^8]^2 \end{aligned}$$

3) META-ANÁLISE

3.1) MODELO COM TAXA DE PENETRÂNCIA CONSTANTE

Reunindo os dados de todas as 21 famílias (incluindo a família estudada em nosso laboratório), obtemos a seguinte função de verossimilhança:

$$\begin{aligned} P = & K^{148} \cdot (1-K)^{121} \cdot (2-K)^{200} \cdot (3-3K+K^2)^2 \cdot [2+(1-K)(2-K)]^7 \\ & \cdot [4+(1-K)(2-K)^2]^{16} \cdot [8+(1-K)(2-K)^3]^7 \cdot [16+(1-K)(2-K)^4]^3 \\ & \cdot [32+(1-K)(2-K)^5]^5 \cdot [64+(1-K)(2-K)^6]^5 \cdot [128+(1-K)(2-K)^7] \\ & \cdot [512+(1-K)(2-K)^9]^2 \cdot [18+9(1-K)(2-K)^2+2(1-K)^2(3-2K)^2] \\ & \cdot [2304+576(1-K)(2-K)^2+9(1-K)^2(2-K)^8+2(1-K)^3(2-K)^6(3-2K)^2] \\ & \cdot [1458(2-K)^4+5832(2-K)^2+64(1-K)^2(3-2K)^5+729(1-K)(2-K)^5] \\ & \cdot [6+3(1-K)+3(1-K)^2+2(1-K)^3] \cdot \{243\{512+(1-K)(2-K)[2+(1-K)(2-K)] \\ & \cdot [4+(1-K)(2-K)^2]^2\} [8+(1-K)(2-K)^3] \{16+(1-K)[8+(1-K)(2-K)^3]\} \\ & + 512(1-K)\{256+(1-K)(2-K)[2+(1-K)(2-K)] [4+(1-K)(2-K)^2]^2\} \\ & \cdot [4+(1-K)(2-K)^3]\{8+(1-K)[8+(1-K)(2-K)^3]\}\} \cdot \{6561(2-K)^5 \\ & \cdot [2+(1-K)+(1-K)^2]+1024(3-2K)^5(1-K)^2[1+(1-K)]\} \\ & \cdot \{(2-K)^2[729+243(1-K)]+(1-K)(3-2K)[324(1-K)(2-K) \\ & + 216 \cdot (2-K)+96(1-K)^2(3-2K)]\} \end{aligned}$$

A estimativa de máxima verossimilhança para a taxa de penetrância global é $K = 0,392$, com um intervalo exato de confiança a 95% de $0,339$ a $0,447$.

O gráfico a seguir, obtido através de código Mathematica, apresenta a função de verossimilhança acima, bem como os limites do intervalo exato de confiança correspondente à estimativa $K = 0,392$.

3.2) MODELO COM TAXA DE PENETRÂNCIA VARIÁVEL

Reunindo-se apenas os dados das 13 genealogias que apresentam mais de três gerações, obtivemos as estimativas de máxima verossimilhança para as taxas globais de penetrância (**K₁** e **K₂**), a partir de funções de verossimilhança construídas sob quatro agrupamentos distintos:

- (1) primeira geração com **K₁** e demais com **K₂**;
- (2) duas primeiras gerações com **K₁** e demais com **K₂**;
- (3) três primeiras gerações com **K₁** e demais com **K₂**;
- (4) quatro primeiras gerações com **K₁** e demais com **K₂**.

Nessas situações, as estimativas **K₁** e **K₂** apresentam os seguintes valores:

Agrupamento	K₁	K₂
1	0,143	0,406
2	0,169	0,455
3	0,283	0,474
4	0,346	0,490

Além da heterogeneidade natural existente entre as genealogias incluídas na meta-análise, a ocorrência de um número variável de gerações que tiveram que ser aglutinadas para constituir os agrupamentos (1) a (4) complicou ainda mais essa situação, inviabilizando a aplicação de testes G (como foi feito nas genealogias isoladas). De fato, os valores obtidos com esses testes foram, como era de se esperar, inapropriados. Da maneira relativamente simplificada como nossos dados foram tratados, não há como se aplicar um teste estatístico adequado ao conjunto usado na meta-análise.

V) PROGRAMAS COMPUTACIONAIS

Com o objetivo de otimizar a estimativa da taxa de penetrância (K) associada a doenças e defeitos com modo de herança autossômico dominante, desenvolvemos o programa **PenCalc**, o qual estima o valor de K com base nas informações contidas nas genealogias, fornecendo o intervalo exato de confiança a 95% dessa estimativa e a fórmula da função de verossimilhança da associada ao heredograma estudado. Os códigos de programação foram desenvolvidos utilizando a linguagem Microsoft® Visual Basic® 6.0, sendo os algoritmos baseados nos métodos descritos no capítulo III.

O programa e seu manual de instruções são apresentados em língua inglesa. Descrevemos, a seguir, o manual traduzido.

1) PENCALC V. 1.0: MANUAL DE INSTRUÇÕES

A figura abaixo mostra a tela de abertura do programa. Esta e as demais figuras do aplicativo foram obtidas de imagens do programa sendo executado em tempo real.

O menu principal da tela de abertura apresenta quatro opções, cada uma delas com os seguintes submenus:

Main Menu	Submenus	
File	Save as...	
	Print	
	Exit	
Data Input	File	Create
		Open
	Screen	
Help	Manual	
	About...	

As opções **Save as...** e **Print** do menu **File** permanecem desabilitadas até que os resultados (fórmula da função de verossimilhança, valor da estimativa de K e intervalo exato de confiança a 95% de K) sejam mostrados na tela. Ambas as opções acessam janelas padrão do Windows® para gravação de arquivos e impressão de dados, respectivamente. A opção **Exit** permite fechar o programa.

DATA INPUT

O programa permite a entrada de dados (1) através de um arquivo de formato pré-definido que é criado usando a opção **Create** do submenu **File**; (2) acessando um arquivo de dados existente através da opção **Open** do submenu **File**; (3) informando os dados através de telas específicas do programa, selecionando a opção **Screen**.

1) CREATE

Essa opção acessa uma caixa de texto para edição de dados, como mostra a tela abaixo:

Para que o arquivo possa ser lido corretamente, os dados correspondentes às estruturas incluídas na genealogia devem seguir rigidamente um formato padronizado pré-estabelecido.

Por exemplo, dada a genealogia com as seguintes estruturas:

o arquivo de dados deverá apresentar o seguinte formato:

```
5,4,2
2,2,3
3,4,0,2,3,5,2,1,7,3,0,2,5
1,3,2,1,2,0,2,3,0
0
0,0,0
0
```

Os dados na primeira linha **(5,4,2)** indicam os números de indivíduos heterozigotos penetrantes (**◆**), heterozigotos não penetrantes (**◈**) e normais sem filhos, descendentes de portador certo do gene(**◇**), respectivamente.

Na segunda linha **(2,2,3)**, o primeiro dígito **(2)** corresponde ao número de árvores de indivíduos normais com duas gerações; os dígitos seguintes **(2,3)** são seus números

correspondentes de filhos. Nesse exemplo, há duas árvores de duas gerações com dois e três indivíduos na segunda geração.

O primeiro dígito **(3)** na terceira linha **(3,4,0,2,3,5,2,1,7,3,0,2,5)** representa o número de árvores de indivíduos normais com três gerações; os dígitos seguintes indicam, em sequência, o número de indivíduos na segunda geração com seus respectivos números de filhos. Nessa genealogia, existem três árvores normais de três gerações: a primeira delas **(4,0,2,3,5)** apresenta 4 indivíduos na segunda geração com, respectivamente, 0, 2, 3 e 5 filhos; a segunda árvore **(2,1,7)** tem 2 indivíduos na segunda geração com, respectivamente, 1 e 7 filhos; e a terceira **(3,0,2,5)** tem 3 indivíduos na segunda geração, respectivamente, com 0, 2 e 5 filhos.

A sequência **(1,3,2,1,2,0,2,3,0)** na quarta linha indica o número de árvores de indivíduos normais com quatro gerações (no exemplo, apenas uma), seguido pelos números correspondentes de árvores de três gerações. Nesta genealogia, existem três árvores de três gerações, a primeira **(2,1,2)** com duas árvores de duas gerações (com um número correspondente de 1 e 2 filhos), a segunda **(0)** sem nenhuma árvore de duas gerações (naturalmente, uma árvore de três gerações sem nenhuma árvore de duas gerações associada a ela se reduz ao caso de uma árvore de uma geração) e a terceira **(2,3,0)** com duas árvores de duas gerações com 3 e 0 filhos, respectivamente.

A próxima linha contém um zero **(0)**, indicando que não há árvores de indivíduos normais de cinco gerações na genealogia. Quando este número é diferente de zero, a construção do arquivo segue as regras válidas para as árvores com um número menor de gerações.

As quantidades **(0,0,0)** na sexta linha representam, respectivamente, o número de pares de gêmeos de mesmo sexo com ambos indivíduos normais, um normal e um afetado e ambos afetados.

Finalmente, o número **0** na última linha do arquivo indica que não há árvores com uniões consanguíneas na genealogia. Este caso será considerado em detalhes no exemplo seguinte, que utiliza uma família descrita por Majewski et al.

(1985), com casos da síndrome da ectrodactilia associada à hemimelia tibial:

O arquivo de dados que representa essa genealogia tem o seguinte formato:

```

13,6,31
2,6,2
0
0
0
0,0,2
2
0,0,1
1,2
0
0
0
0,0,0
99
0,0,0
2,2,7
0
0
0
0,0,0
0,0,1
0
1,7,2,0,0,0,0,0,0
0
0
0,0,0

```

Os números na primeira linha **(13,6,31)** indicam que existem 13 indivíduos heterozigotos penetrantes (**◆**), 6

heterozigotos não penetrantes ($\diamond$) e 31 indivíduos normais sem filhos, descendentes de portador certo do gene($\diamond$).

Os elementos na segunda linha **(2,6,2)** representam as duas árvores de indivíduos normais de duas gerações correspondendo aos indivíduos (IV.1 e V.1-6) com 6 indivíduos na segunda geração e (IV.7 e V.9-10) com 2 indivíduos na segunda geração.

Os números **0** na terceira, quarta e quinta linhas indicam que não existem árvores de indivíduos normais de três, quatro e cinco gerações na genealogia.

A próxima linha do arquivo de dados **(0,0,2)** representa os números de pares gemelares de mesmo sexo com ambos indivíduos normais, um normal e um afetado e ambos afetados. A genealogia contém dois pares gemelares (indivíduos IV.21-22 e IV.40-41), sendo ambos do sexo feminino e afetados.

Na sexta linha, o número **2** indica o número de árvores com uniões consanguíneas existente na genealogia.

A primeira árvore é representada pelos indivíduos IV.9, IV.20 e V.11-12. Levando em conta que o gene patológico, se presente nesta árvore, pode ser transmitido aos indivíduos V.11 ou V.12 através do indivíduo IV.9 ou IV.20 (considerando desprezível, para a proposta específica de construção da função de verossimilhança, a probabilidade de transmissão através de ambos indivíduos), desfazemos a união consanguínea em duas possíveis configurações não consanguíneas:

As próximas seis linhas do arquivo contêm os elementos pertinentes à primeira configuração possível (acima à esquerda).

A primeira linha **(0,0,1)** descreve os números de indivíduos heterozigotos penetrantes ($\blacklozenge$), heterozigotos não

penetrantes ($\diamond$) e indivíduos normais sem filhos, descendentes de portador certo do gene ($\diamond$), respectivamente, 0, 0 e 1.

A segunda linha **(1,2)** lista o número de árvores de indivíduos normais com duas gerações (no exemplo, 1) e o número correspondente de indivíduos na segunda geração (2).

As três linhas seguintes apresentam um **0**, indicando que não existem árvores de indivíduos normais de três, quatro e cinco gerações nesta configuração.

Finalmente, a última linha **(0,0,0)** mostra que não há ocorrência de pares gemelares de mesmo sexo de quaisquer tipos.

O arquivo apresenta, então, os dados correspondentes à segunda configuração possível para a primeira árvore com união consanguínea. Essa estrutura é exatamente a mesma da primeira configuração. O número **99** da linha seguinte indica ao programa que existe mais um conjunto de dados semelhante ao previamente informado, o que equivale à duplicação das linhas referentes à primeira configuração consanguínea.

A segunda árvore com união consanguínea é representada pelos indivíduos III.5, IV.25-32 e V.15-16. Desfazendo-se a união consanguínea, obtemos duas configurações possíveis:

As próximas seis linhas do arquivo apresentam os elementos pertinentes à primeira configuração (acima à esquerda).

A primeira linha **(0,0,0)** indica que não há nenhum indivíduo heterozigoto penetrante ($\blacklozenge$), heterozigoto não penetrante ($\diamond$) ou normal sem filhos, descendente de portador certo do gene ($\diamond$).

A segunda linha **(2,2,7)** lista os números de árvores de indivíduos normais com duas gerações (no exemplo, 2) e seus correspondentes números de indivíduos na segunda geração (2 e 7).

Os números **0** nas três linhas seguintes indicam que não há árvores de indivíduos normais com três, quatro e cinco gerações nesta configuração.

Finalmente, a última linha **(0,0,0)** lista os números de pares gemelares de mesmo sexo com ambos indivíduos normais, um normal e um afetado e ambos afetados.

A configuração alternativa desta árvore (acima à direita) é descrita pelos elementos nas próximas seis linhas do arquivo.

A primeira linha **(0,0,1)** descreve que não há nenhum indivíduo heterozigoto penetrante ($\blacklozenge$) ou não penetrante ($\blacklozenge$) e apenas um indivíduo normal sem filhos, descendente de portador certo do gene($\blacklozenge$).

Na segunda linha, o número **0** indica que não há árvores de indivíduos normais de duas gerações.

O primeiro dígito **(1,7,2,0,0,0,0,0,0)** da terceira linha **(1)** representa o número de árvores de indivíduos normais de três gerações; os dígitos seguintes, descrevem, em sequência, os números de indivíduos na segunda geração e seus correspondentes números de filhos. No exemplo acima, há apenas uma árvore de três gerações com 7 indivíduos na segunda geração, o primeiro deles com 2 filhos e os demais, sem filhos (indicado nesta linha do arquivo por seis números zero).

Os números **0** nas próximas duas linhas indicam que não há árvores de indivíduos normais com quatro e cinco gerações.

Finalmente, a última linha **(0,0,0)** mostra que não há ocorrência de pares gemelares de mesmo sexo de quaisquer tipos.

Uma vez digitados, os dados devem ser gravados num arquivo utilizando o botão **SAVE AS...** dessa tela, o qual abre uma janela padrão do Windows® para gravação dos dados. O programa salva as informações no arquivo e automaticamente exibe os resultados na tela.

2) OPEN

Ao acessar a opção **Open** do submenu **File**, o programa abre uma janela padrão Windows® para localização de arquivos. Carregando o arquivo através do botão de comando **OPEN**, o programa automaticamente lê as informações e exibe os resultados na tela.

3) SCREEN

Ao selecionar a opção **Screen**, o programa apresenta a tela principal de entrada de dados:

The screenshot shows the 'PenCalc 1.0' window with a title bar containing standard Windows controls. The main area is titled 'Numbers of observed configurations'. It features a grid of nine boxes, each containing a different configuration of diamond shapes. Below the grid are three empty text input boxes, followed by six 'OK' buttons, and finally 'Back' and 'Enter' buttons at the bottom right.

Numbers of observed configurations								
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="OK"/>	<input type="button" value="OK"/>	<input type="button" value="OK"/>	<input type="button" value="OK"/>	<input type="button" value="OK"/>	<input type="button" value="OK"/>
<input type="button" value="Back"/> <input type="button" value="Enter"/>								

As três caixas de texto à esquerda devem ser completadas, respectivamente, com os números de indivíduos heterozigotos penetrantes, heterozigotos não penetrantes e normais sem filhos, descendentes de um portador certo do gene.

Os botões de comando nessa interface gráfica executam as seguintes tarefas:

OK acessam as telas para entrada de dados de árvores de indivíduos normais com duas, três, quatro e cinco gerações; de árvores com uniões consanguíneas; e de pares gemelares.

Back retorna à tela principal, sem gravar os dados digitados.

Enter retorna à tela principal ou acessa as telas para entrada de dados de outras árvores (no caso de mais de uma árvore com quatro ou cinco gerações), sempre gravando os dados digitados.

3.1) ÁRVORES DE INDIVÍDUOS NORMAIS COM DUAS GERAÇÕES

Na caixa de texto, deve-se indicar o número de árvores de indivíduos normais com duas gerações (máximo: 15).

O botão de comando **OK** libera as caixas de texto correspondentes ao número de árvores para entrada do número de indivíduos da segunda geração em cada uma delas, como mostra a tela abaixo:

PenCalc 1.0

No. of 2-generation trees

No. of individuals in the second generation

Tree 1	Tree 2	Tree 3	Tree 4	Tree 5
<input type="text" value="1"/>	<input type="text" value="3"/>	<input type="text" value="5"/>	<input type="text" value="2"/>	<input type="text" value="10"/>
Tree 6	Tree 7	Tree 8	Tree 9	Tree 10
<input type="text" value="3"/>	<input type="text" value="3"/>	<input type="text" value="8"/>	<input type="text" value="0"/>	<input type="text" value="1"/>
Tree 11	Tree 12	Tree 13	Tree 14	Tree 15
<input type="text" value="1"/>	<input type="text" value="9"/>	<input type="text" value="7"/>	<input type="text" value="2"/>	<input type="text" value="1"/>

3.2) ÁRVORES DE INDIVÍDUOS NORMAIS COM TRÊS GERAÇÕES

Como anteriormente, uma caixa de texto é utilizada para digitar o número de árvores (máximo: 10).

O botão de comando **OK** abre um novo conjunto de caixas de texto permitindo a entrada de dados dos indivíduos da segunda geração. Para liberar as caixas referentes à terceira geração, o usuário deve digitar o número de indivíduos existente na segunda geração e teclar **ENTER**, como no exemplo abaixo:

PenCalc 1.0

No. of 3-generation trees

Tree 1 - No. of individuals in the second generation

Offspring no. of each individual

Tree 2 - No. of individuals in the second generation

Tree 3 - No. of individuals in the second generation

Quando todos os dados forem digitados (como no exemplo abaixo), o botão de comando **ENTER** deve ser pressionado para retornar o programa à tela inicial:

The screenshot shows the 'PenCalc 1.0' window. At the top, it says 'No. of 3-generation trees' with a text box containing '4', and 'OK' and 'Back' buttons. Below this, there are three sections for tree data entry. Each section has a label 'Tree X - No. of individuals in the second generation', a text box for the number of individuals, and a label 'Offspring no. of each individual' followed by a row of text boxes for the offspring counts. An 'Enter' button is located to the right of the first tree's data entry fields.

Tree	No. of individuals in the second generation	Offspring no. of each individual
Tree 1	2	2, 3
Tree 2	5	1, 0, 4, 3, 3
Tree 3	10	1, 2, 5, 7, 4, 1, 0, 0, 3, 3

O número máximo de indivíduos na segunda geração é dez; na terceira geração, não há restrições quanto a esse número.

O acesso ao restante da tela é feito através da barra de rolagem, que só é disponibilizada caso o número de árvores seja superior a três.

3.3) ÁRVORES DE INDIVÍDUOS NORMAIS COM QUATRO GERAÇÕES

Após digitar o número de árvores (máximo: 10) e confirmar a entrada do dado com o botão **OK**, surge a caixa de texto para entrada do número de indivíduos existente na segunda geração. A tecla **ENTER** deverá ser utilizada para liberar as caixas de texto de entrada do número de filhos de cada indivíduo na segunda e terceira gerações, como mostra o exemplo abaixo:

PenCalc 1.0

No. of 4-generation trees

Tree 1 - No. of individuals in the second generation

Individual 1 - offspring no.
 Offspring no. of each individual in the third generation

Individual 2 - offspring no.
 Offspring no. of each individual in the third generation

Individual 3 - offspring no.
 Offspring no. of each individual in the third generation

O número máximo de indivíduos na segunda e terceira gerações é dez; na quarta geração, não há restrições quanto a esse número. Um exemplo da tela já preenchida com os dados da primeira árvore de quatro gerações é mostrado abaixo; a barra de rolagem deverá ser utilizada para acessar os dados correspondentes ao quarto indivíduo. Para gravar esses dados e acessar a interface gráfica usada para receber os dados da segunda árvore de quatro gerações, o botão de comando **ENTER** deve ser pressionado.

PenCalc 1.0

No. of 4-generation trees

Tree 1 - No. of individuals in the second generation

Individual 1 - offspring no.
 Offspring no. of each individual in the third generation

Individual 2 - offspring no.
 Offspring no. of each individual in the third generation

Individual 3 - offspring no.
 Offspring no. of each individual in the third generation

3.4) ÁRVORES DE INDIVÍDUOS NORMAIS COM CINCO GERAÇÕES

O programa aceita, no máximo, cinco árvores de indivíduos normais com cinco gerações. O procedimento é o mesmo observado para os casos de duas a quatro gerações; informa-se,

primeiramente, o número de árvores e o programa libera, sequencialmente, as caixas de texto para entrada dos demais dados, sempre através da tecla **ENTER**. Abaixo, um exemplo do formato da tela:

PenCalc 1.0

No. of 5-generation trees: **OK** **Back** **Enter**

Tree 1 - No. of individuals in the second generation

Individual 1 (second generation) - offspring no.

Individual 1 - offspring no.	<input type="text" value="2"/>	Offspring no.	<input type="text" value="3"/>	<input type="text" value="3"/>
Individual 2 - offspring no.	<input type="text" value="5"/>	Offspring no.	<input type="text" value="1"/>	<input type="text" value="0"/>
Individual 3 - offspring no.	<input type="text" value="3"/>	Offspring no.	<input type="text" value="2"/>	<input type="text" value="4"/>
Individual 4 - offspring no.	<input type="text" value="1"/>	Offspring no.	<input type="text" value="2"/>	<input type="text" value="8"/>
Individual 5 - offspring no.	<input type="text" value="6"/>	Offspring no.	<input type="text" value="1"/>	<input type="text" value="0"/>
Individual 6 - offspring no.	<input type="text" value="4"/>	Offspring no.	<input type="text" value="3"/>	<input type="text" value="2"/>

Offspring no. | | | | |

Offspring no. | | |

O programa aceita, no máximo, seis indivíduos por geração, com exceção da quinta geração, onde esse número é ilimitado, como nos casos anteriores. As barras de rolagem horizontal e vertical devem ser utilizadas para acessar outras partes da tela.

3.5) ÁRVORES COM UNIÕES CONSANGUÍNEAS

Após digitar o número de árvores com uniões consanguíneas (máximo: 2), o programa libera as caixas de texto para entrada das informações pertinentes a cada configuração. Um exemplo é mostrado abaixo, o qual diz respeito às configurações possíveis da árvore

+

configuração 1

+

configuração 2

O procedimento detalhado para entrada de dados de árvores com uniões consanguíneas está descrito na seção **MÉTODOS** (do manual em inglês que acompanha o programa).

PenCalc 1.0

No. of consanguineous trees

Consanguineous Tree 1 - Nos. of observed configurations

Configuration 1

<input type="text" value="1"/>		<input type="text"/>		<input type="text" value="1"/>		<input type="text"/>		<input type="text"/>		<input type="button" value="OK"/>
<input type="text" value="4"/>
<input type="text" value="1"/>										

Is configuration 2 different? ☒ Yes ☐ No

Configuration 2

<input type="text" value="1"/>		<input type="text"/>		<input type="text"/>		<input type="text"/>		<input type="text"/>		<input type="button" value="OK"/>
<input type="text" value="6"/>
<input type="text" value="2"/>										

Como anteriormente, as caixas de texto e os botões de comando **OK** acessam as telas correspondentes a cada estrutura.

No exemplo acima, as duas configurações possíveis da árvore, após desfeita a união consanguínea, são diferentes. Caso as duas configurações sejam semelhantes, o programa completa automaticamente os campos, considerando os dados já informados para a primeira delas.

3.6) PARES GEMELARES

Três tipos de pares de gêmeos de mesmo sexo (mono ou dizigóticos) podem ser identificados.

Os números de pares gemelares de cada tipo (ambos indivíduos normais; um indivíduo normal e o outro afetado; ou ambos afetados) são informados através da seguinte interface gráfica (o exemplo indica que há um par de cada tipo):

RESULTADOS

Após a entrada de todos os dados pertinentes, o programa exibe a estimativa da taxa de penetrância K , seu intervalo exato de confiança a 95% $[95\%CI(K)]$ e a fórmula da função de verossimilhança correspondente. Um campo com a opção para identificar a genealogia também é apresentado nessa interface gráfica.

Como um exemplo, para a genealogia com as seguintes estruturas

a tela de resultados do programa toma a seguinte forma:

Quando a genealogia contém árvores com uniões consanguíneas, o programa também mostra os valores parciais de K usados para estimar o valor final da taxa de penetrância, como

no exemplo a seguir, que utiliza a genealogia adaptada de Naveed et al. (2006):

PenCalc 1.0

File Data Input Help

Identification: Naveed et al. (2006) modif.

Likelihood Function

$$F = K^{19} \cdot (1-K)^{22} \cdot (2-K)^{28} \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\} \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^6\}$$

CONSANGUINEOUS TREE 1

Configuration 1:

$$K^3 \cdot (1-K)^2 \cdot (2-K)^9 \cdot \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\}$$

Configuration 2:

$$K^3 \cdot (1-K)^3 \cdot (2-K)^{11}$$

K11 = 0.347

K12 = 0.345

K21 = 0.340

K22 = 0.338

K = 0.342

95%CI (K) = 0.233 - 0.468

A genealogia apresenta duas árvores consanguíneas distintas. Cada uma delas pode ser separada em duas árvores de indivíduos não consanguíneos. As fórmulas completas para as quatro funções de verossimilhança correspondentes a essa genealogia, consistem de uma parte comum (que inclui as árvores não consanguíneas), identificada pela letra **F** abaixo, multiplicada por cada uma das quatro possíveis combinações entre as fórmulas de uma das configurações da primeira árvore consanguínea (f_{11} e f_{12}) e uma (f_{21} e f_{22}) da outra árvore.

$$\begin{aligned} \mathbf{F} = & K^{19} \cdot (1-K)^{22} \cdot (2-K)^{28} \cdot \\ & \{1/2 + (1-K)/2 \cdot [(2-K)/2]^3\} \cdot \\ & \{1/2 + (1-K)/2 \cdot [(2-K)/2]^6\} \cdot \end{aligned}$$

$$\begin{aligned} f_{11} = & K^3 \cdot (1-K)^2 \cdot (2-K)^9 \cdot \\ & \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\} \cdot \\ f_{12} = & K^3 \cdot (1-K)^3 \cdot (2-K)^{11} \cdot \\ f_{21} = & K^1 \cdot (1-K)^2 \cdot (2-K)^1 \cdot \\ & \{1/2 + (1-K)/2 \cdot [(2-K)/2]^2\} \cdot \\ f_{22} = & K^1 \cdot (1-K)^2 \cdot (2-K)^2 \cdot \\ & \{1/2 + (1-K)/2 \cdot [(2-K)/2]^1\} \cdot \end{aligned}$$

ou seja, as quatro funções de verossimilhança possíveis são

$$\begin{aligned} \mathbf{F}_1 &= \mathbf{F} \cdot f_{11} \cdot f_{21} \\ \mathbf{F}_2 &= \mathbf{F} \cdot f_{11} \cdot f_{22} \\ \mathbf{F}_3 &= \mathbf{F} \cdot f_{12} \cdot f_{21} \\ \mathbf{F}_4 &= \mathbf{F} \cdot f_{12} \cdot f_{22} \end{aligned}$$

Como indicado anteriormente, os detalhes sobre a estimativa do valor de **K** para genealogias contendo árvores com uniões consanguíneas são fornecidos na seção **MÉTODOS** (do manual em inglês que acompanha o programa).

Os resultados mostrados na interface gráfica acima podem ser gravados em um arquivo de dados ou impressos usando as opções **Save as...** ou **Print** do menu **File**.

HELP

MANUAL

Exibe uma mensagem indicando o manual de instruções encontra-se no arquivo *manual.pdf*, localizado no diretório de onde o programa foi instalado.

ABOUT

Apresenta os créditos do programa e o endereço profissional dos autores:

A listagem do código do programa encontra-se no **ANEXO II**. O programa executável (versão 1.0 - beta) gravado em disco CD-ROM encontra-se também em anexo no presente trabalho.

2) **PENCALC WEB V. 1.0: UMA VERSÃO DO PROGRAMA PARA INTERNET**

O aluno Márcio Teruo Onodera, graduando em Ciências Biológicas (Instituto de Biociências / USP) desenvolveu em seu projeto de conclusão de curso, sob orientação do Prof. Dr. Paulo A. Otto e com a nossa colaboração, uma versão do programa **PenCalc v. 1.0** para Internet utilizando a estrutura de programação Active Server Pages - ASP, por meio das linguagens VBScript e JScript processadas pelo lado servidor para geração do conteúdo dinâmico apresentado na linguagem HTML na Internet - World Wide Web. Essa versão é executada diretamente em página da web, que pode ser acessada livre e gratuitamente através do endereço <http://www.ib.usp.br/~otto/pencalcweb>.

O aplicativo apresenta uma série de formulários com apoio gráfico que coletam as frequências absolutas das estruturas da genealogia. O processamento dos dados ocorre *on line* e os seguintes resultados são apresentados em uma página com interface gráfica: (a) listagem das estruturas correspondentes à genealogia; (b) estimativa do cálculo da taxa de penetrância K; (c) intervalo exato de confiança a 95% da estimativa; (c) probabilidade de heterozigose de todos os indivíduos da genealogia filtrada; (d) risco de prole afetada de para cada indivíduo.

As Figuras 1 a 5 ilustram as interfaces gráficas do aplicativo referentes à entrada de dados e apresentação de resultados de uma genealogia composta pelas seguintes estruturas:

Affected:

Affected person: a penetrant heterozygote, resulting from a crossing between a heterozygote and a normal homozygote. The corresponding probability is $K/2$ (K for the individual that generates the whole genealogy).

Normal Obligate Carriers:

Normal Obligate Carrier: a known non-penetrant heterozygote, who has a normal phenotype and at least two affected descendants, or one ascendant and one descendant, both affected. The corresponding probability is $(1-K)/2$ ($1-K$ for the individual that generates the whole genealogy).

Normal Offspring of Obligate Carriers:

Normal Offspring of Obligate Carriers: a normal homozygote or non-penetrant heterozygote without offspring, resulting from a crossing between a heterozygote and a normal homozygote. The corresponding probability is $1/2 + (1-K)/2 = (2-K)/2$.

Description: Describe briefly this genealogy, your contacts and other useful information.

Figura 1 - Formulário de entrada de dados referentes a heterozigotos penetrantes, heterozigotos não-penetrantes e indivíduos normais, sem filhos, descendentes de portadores certos do gene.

Step 3 of 3 - Updating 2G trees of normal descendants of obligate carrier

Number of 2-generation trees: 2

Tree 1 - No. of individuals in the 2nd generation: 2
 Tree 2 - No. of individuals in the 2nd generation: 3

2

3

Figura 2 - Formulário de entrada de dados referentes a árvores de indivíduos normais com duas gerações.

Step 4 of 4 - Updating 3G trees of normal descendants of obligate carrier

Number of 3-generation trees: 3

Tree 1 - No. of individuals in the 2nd generation: 4
 Number of offspring for each individual: 0 2 3 5
 Tree 2 - No. of individuals in the 2nd generation: 2
 Number of offspring for each individual: 1 7
 Tree 3 - No. of individuals in the 2nd generation: 3
 Number of offspring for each individual: 0 2 5

2 3 5

1 7

2 5

Figura 3 - Formulário de entrada de dados referentes a árvores de indivíduos normais com três gerações.

Figura 5 - Página de resultado com uma interface gráfica apresentando (a) as estruturas correspondentes aos componentes da genealogia; (b) o valor da estimativa da taxa de penetrância e seu respectivo intervalo exato de confiança a 95%; (c) a probabilidade de heterozigose e o risco de prole afetada de todos os indivíduos da genealogia filtrada.

Em todos os módulos, os algoritmos seguem rigorosamente os métodos definidos no capítulo III deste trabalho.

Na Tabela abaixo, apresentamos um painel comparativo das operações realizadas por cada versão (executável e Internet) do programa **PenCalc v. 1.0**, visto que os diferentes módulos não são completamente equivalentes entre elas.

Tabela 21 - Módulos existentes nas versões (executável e Internet) do programa PenCalc v. 1.0.

Módulos	PenCalc v. 1.0	
	Executável	Internet
Entrada de dados		
através de arquivo	X	
através das telas	X	X
árvores de indivíduos normais até cinco gerações	X	X
árvores com consaguinidade	X	X
gêmeos	X	X
Resultados		
estimativa da taxa de penetrância K	X	X
intervalo exato de confiança da estimativa de K [IC95%(K)]	X	X
fórmula da função de verossimilhança da genealogia	X	
desenho das estruturas da genealogia		X
probabilidade de heterozigose de indivíduos normais		X
cálculo de risco de recorrência por indivíduo		X

Em razão das dificuldades em desenvolver na versão executável, usando a linguagem Microsoft® Visual Basic® 6.0, os módulos de cálculo de probabilidade de heterozigose e risco de afecção na prole para todos os indivíduos da genealogia, optamos por desenvolvê-los apenas na versão do programa para Internet, cuja linguagem de programação oferece uma interface gráfica mais intuitiva ao usuário.

O programa **PenCalc v. 1.0** apresenta-se em versão beta, ou seja, encontra-se disponível e em operação, ainda que esteja em fase de desenvolvimento e testes, aberto para coletar impressões e sugestões de usuários e para correção de eventuais falhas.

VI) RESUMO

O objetivo principal do trabalho foi o desenvolvimento de um programa computacional, em linguagem Microsoft® Visual Basic® 6.0 (versão executável), para estimativa da taxa de penetrância a partir da análise de genealogias com casos de doenças com herança autossômica dominante.

Embora muitos dos algoritmos empregados no programa tenham se baseado em idéias já publicadas na literatura (em sua maioria por pesquisadores e pós-graduandos do Laboratório de Genética Humana do Instituto de Biociências da Universidade de São Paulo), desenvolvemos alguns métodos inéditos para lidar com situações encontradas com certa frequência nos heredogramas publicados na literatura, como: a) ausência de informações sobre o fenótipo do indivíduo gerador da genealogia; b) agrupamento de árvores de indivíduos normais sem a descrição da distribuição de filhos entre os progenitores; c) análise de estruturas da genealogia contendo uniões consanguíneas, utilizando um método alternativo ao descrito na literatura; d) determinação de soluções gerais para as funções de verossimilhança de árvores de indivíduos normais com ramificação regular e para as probabilidades de heterozigose de qualquer indivíduo pertencente a essas árvores.

Além da versão executável, o programa, denominado *PenCalc*, é apresentado também numa versão para Internet (*PenCalc Web*), a qual fornece adicionalmente as probabilidades de heterozigose e o cálculo de afecção na prole de todos os indivíduos da genealogia. Essa versão pode ser acessada livre e gratuitamente no endereço <http://www.ib.usp.br/~otto/pencalcweb>.

Desenvolvemos também um modelo com taxa de penetrância variável dependente da geração, uma vez que a inspeção de famílias com doenças autossômicas dominantes, como é o caso da síndrome da ectrodactilia associada à hemimelia tibial (EHT), sugere a existência de um fenômeno similar à antecipação, em relação à taxa de penetrância. Os modelos com taxa de penetrância constante e variável, e os métodos desenvolvidos

neste trabalho foram aplicados a 21 heredogramas de famílias com afetados pela EHT e ao conjunto das informações de todas essas genealogias (meta-análise), obtendo-se em todos os casos estimativas da taxa de penetrância.

VII) ABSTRACT

The main objective of this dissertation was the development of a computer program, in Microsoft® Visual Basic® 6.0, for estimating the penetrance rate of autosomal dominant diseases by means of the information contained on genealogies.

Some of the algorithms we used in the program were based on ideas already published in the literature by researchers and (post-) graduate students of the Laboratory of Human Genetics, Department of Genetics and Evolutionary Biology, Institute of Biosciences, University of São Paulo. We developed several other methods to deal with particular structures found frequently in the genealogies published in the literature, such as: a) the absence of information on the phenotype of the individual generating of the genealogy; b) the grouping of trees of normal individuals without the separate description of the offspring number per individual; c) the analysis of structures containing consanguineous unions; d) the determination of general solutions in simple analytic form for the likelihood functions of trees of normal individuals with regular branching and for the heterozygosis probabilities of any individual belonging to these trees.

In addition to the executable version of the program summarized above, we also prepared, in collaboration with the dissertation supervisor and the undergraduate student Marcio T. Onodera (main author of this particular version), another program, represented by a web version (PenCalc Web). It enables the calculation of heterozygosis probabilities and the offspring risk for all individuals of the genealogy, two details we did not include in the present version of our program. The program PenCalc Web can be accessed freely at the home-page address <http://www.ib.usp.br/~otto/pencalcweb>.

Another important contribution of this dissertation was the development of a model of estimation with generation-dependent penetrance rate, as suggested by the inspection of families with some autosomal dominant diseases, such as the ectrodactyly-tibial hemimelia syndrome (ETH), a condition which

exhibits a phenomenon similar to anticipation in relation to the penetrance rate. The models with constant and variable penetrance rates, as well as practically all the methods developed in this dissertation, were applied to 21 individual genealogies from the literature with cases of ETH and to the set of all these genealogies (meta-analysis). The corresponding results of all these analysis are comprehensively presented.

VIII) REFERÊNCIAS BIBLIOGRÁFICAS

AHMAD M et al. X-chromosomally inherited split-hand/split-foot anomaly in a Pakistani kindred. **Hum. Genet.** 75: 169-173, 1987.

ALLEN G. The meaning of concordance and discordance in estimation of penetrance and gene frequency. **Am. J. Hum. Genet.** 4(3): 155-172, 1952.

BADANO JL, KATSANIS N. Beyond Mendel: an evolving view of human genetic disease transmission. **Nature Rev. Genet.** 3: 779-789, 2002.

BERGLIN CG. On the concepts and calculus of penetrance. **Acta Genet. Med. Stat.** 5: 240-262, 1955.

BERGLIN CG. Some penetrance formulae in recessive proband material. **Acta Genet. Med. Gemellol.** 6: 451-458, 1957.

BOLK S et al. A human model for multigenic inheritance: phenotypic expression in Hirschsprung disease requires both the RET gene and a new 9q31 locus. **Proc. Nat. Acad. Sci.** 97(1): 268-273, 2000.

BOLK S et al. Segregation at three loci explains familial and population risk in Hirschsprung disease. **Nature Genet.** 31: 89-93, 2002.

BRUDER CEG et al. Phenotypically concordant and discordant monozygotic twins display different DNA copy-number-variation profiles. **Am. J. Hum. Genet.** 82: 763-771, 2008.

CANNINGS C, THOMPSON EA, SKOLNICK M. In: MIELKE JH & CRAWFORD MH. **Current developments in anthropological genetics**. New York: Plenum Press, 1980.

CUMMINGS CJ, ZOGHBI HY. Fourteen and counting: unraveling trinucleotide repeat diseases. **Hum. Mol. Genet.** 9(6): 909-916, 2000.

DAYER R et al. Tibial aplasia-hypoplasia and ectrodactyly in monozygotic twins with a discordant phenotype. **J. Pediatr. Orthop.** 27(3): 266-269, 2007.

ELANDT-JOHNSON RC. Segregation analysis for complex modes of inheritance. **Am. J. Hum. Genet.** 22: 129-144, 1970.

FAYAZ-UL-HAQUE M et al. X-chromosomal split-hand/split-foot anomaly maps to Xq26. **Hum. Genet.** 91: 17-19, 1993.

FAYAZ-UL-HAQUE M et al. Fine mapping of the X-linked split-hand/split-foot malformation (SHFM2) locus to a 5.1-Mb region on Xq26.3 and analysis of candidate genes. **Clin. Genet.** 67: 93-97, 2005.

FRAGA MF et al. Epigenetic differences arise during the lifetime of monozygotic twins. **Proc. Nat. Acad. Sci.** 102(30): 10604-10609, 2005.

FROTA-PESSOA O, OTTO PA, OLIVARES-PLAZA JR. The variation of recurrence risks with penetrance for isolated cases of autosomal dominant conditions. **J. Hered.** 67: 256, 1976.

GRINGRAS P, CHEN W. Mechanisms for differences in monozygous twins. **Early Hum. Develop.** 64: 105-117, 2001.

HALL JG. Twinning: mechanisms and genetic implications. **Curr. Opinion Genet. Develop.** 6: 343-347, 1996.

HARBOUR JW. Molecular basis of low-penetrance retinoblastoma. **Arch. Ophthalmol.** 119: 1699-1704, 2001.

HEYNINGEN V, YEYATI PL. Mechanisms of non-Mendelian inheritance in genetic disease. **Hum. Mol. Genet.** 13: 225-233, 2004.

HORIMOTO ARVR, OTTO PA. Penetrance Rate Estimation. In: OTTO PA. **Métodos clásicos y modernos para el análisis de datos en genética humana**. Posadas, Argentina: Editorial Universitaria de la Universidad Nacional de Misiones, 2008. p.123-150. Disponível em: http://www.ib.usp.br/~otto/pop_genetics.htm. Acesso em: 8 jun. 2009.

HOYME HE et al. Autosomal dominant ectrodactyly and absence of long bones of upper or lower limbs: further clinical delineation. **J. Pediatr.** **111**: 538-543, 1987.

KOHN G, SHAWWA R, GRUNEBAUM M. Aplasia of the tibia with bifurcation of the femur and ectrodactyly: evidence for an autosomal recessive type. **Am. J. Med. Genet.** **33**: 172-175, 1989.

KNUDSON AG. Mutation and cancer: statistical study of retinoblastoma. **Proc. Nat. Acad. Sci.** **68(4)**:820-823, 1971.

KNUDSON AG. Genetics of human cancer. **Ann. Rev. Genet.** **20**: 231-251, 1986.

LASKER GW. Penetrance estimated by the frequency of unilateral occurrences and by discordance in monozygotic twins. **Hum. Biol.** **19**: 217-230, 1947.

MAESTRELLI SRP, OTTO PA. Estimativa da penetrância e cálculo de riscos de recorrência em doenças condicionadas por mecanismo hereditário autossômico dominante com penetrância incompleta. **Braz. J. Dysmorphol. Speech-Hearing Dis.** **1**: 55-66. 1997.

MAJEWSKI et al. Aplasia of tibia with split-hand/split-foot deformity. Report of six families with 35 cases and considerations about variability and penetrance. **Hum. Genet.** **70**: 136-147, 1985.

MAJEWSKI F, GOECKE T, MEINECKE P. Ectrodactyly and absence (hypoplasia) of the tibia: are there dominant and recessive types? **Am. J. Med. Genet.** **63**: 185-189, 1996.

MARINONI JC et al. Familial split hand/split foot long bone deficiency does not segregate with markers linked to the SHFD1 locus in 7q21.3-q22.1. **Hum. Mol. Genet.** 3(8): 1355-1357, 1994.

MATSUNAGA E. On estimating penetrance of the retinoblastoma gene. **Hum. Genet.** 56: 127-128, 1980.

MATSUYAMA J et al. A pair of sibs with tibial hemimelia born to phenotypically normal parents. **J. Hum. Genet.** 48: 173-176, 2003.

McGEE TL et al. Evidence that the penetrance of mutations at the RP11 locus causing dominant retinitis pigmentosa is influenced by a gene linked to the homologous RP11 allele. **Am. J. Hum. Genet.** 61: 1059-1066, 1997.

MUFTI MH, WOOD SK. Ectrodactyly in sisters and half sisters. **J. Med. Genet.** 24: 220-224, 1987.

NAVEED M et al. Ectrodactyly with aplasia of long bones (OMIM 119100) in a large inbred arab family with an apparent autosomal dominant inheritance and reduced penetrance: clinical and genetic analysis. **Am. J. Med. Genet.** 140A: 1440-1446, 2006.

NAVEED M et al. Genomewide linkage scan for split-hand/foot malformation with long-bone deficiency in a large arab family identifies two novel susceptibility loci on chromosomes 1q42.2-q43 and 6q14.1. **Am. J. Hum. Genet.** 80: 105-111, 2007.

NITU-WHALLEY IC et al. Type 1 von Willebrand disease - a clinical retrospective study of the diagnosis, the influence of the ABO blood group and the role of the bleeding history. **Br. J. Haematol.** 108: 259-264, 2000.

OTTO PA, FROTA-PESSOA O, POLCAN SP. Snyder's ratios with incomplete penetrance. **J. Hered.** 85(4): 331-335, 1994.

OTTO PA, HORIMOTO ARVR. **Statistical Distributions: exact probability values of the Gaussian, t, F and chi-squared distributions**. São Paulo: Instituto de Biociências, USP. 2006. 334kB. Disponível em: <http://www.ib.usp.br/~otto/software.htm>. Acesso em 10 Jun. 2008.

OTTO PA, MAESTRELLI SRP. Heterozygosity probabilities for normal relatives of isolated cases affected by incompletely penetrant conditions and the calculation of recurrence risks for their offspring. I. Autosomal dominant genes. **Am. J. Med. Genet.** 95: 43-48, 2000.

PASSARGE E. Dissecting Hirschsprung disease. **Nature Genet.** 31: 11-12, 2002.

PASTERNAK, JJ. **Uma introdução à genética molecular humana: mecanismos das doenças hereditárias**. 2.ed. Rio de Janeiro: Guanabara Koogan, 2007.

PATERSON AD et al. Genetic anticipation in neurological and other disorders. In: WELLS RD, WARREN ST. **Genetic instabilities and hereditary neurological diseases**. San Diego, California: Academic Press, 1998. p.413-428.

PFÄNDLER U. La signification statistique de la concordance chez les jumeaux MZ et DZ. **Acta Genet. Med. Gemellol.** 19: 160-164, 1970.

PRAXEDES LA. **Estimativas do valor da penetrância utilizando dados de gêmeos monozigóticos e dizigóticos**. 1997. 176p. Dissertação(Mestrado) - Instituto de Biociências, Universidade de São Paulo, São Paulo, 1997.

PRAXEDES LA, OTTO PA. Estimation of penetrance from twin data. **Twin Res.** 3: 294-298, 2000.

RICHIERI-COSTA et al. Tibial hemimelia: Report on 37 new cases, clinical and genetic considerations. **Am. J. Med. Genet.** 27: 867-884, 1987.

RIVOLTA C et al. Retinitis pigmentosa and allied diseases: numerous diseases, genes, and inheritance patterns. **Hum. Mol. Genet.** 11(10): 1219-1227, 2002.

ROGATKO A. **Solução bayesiana para dois problemas clássicos em genética: penetrância e equilíbrio de Hardy-Weinberg.** 1983. 113p. Tese(Doutorado) - Instituto de Biociências, Universidade de São Paulo, São Paulo, 1983.

ROGATKO A, PEREIRA CAB, FROTA-PESSOA O. A Bayesian method for the estimation of penetrance: application to mandibulofacial and frontonasal dysostoses. **Am. J. Med. Genet.** 24: 231-246, 1986.

SANG, JH. Penetrance, expressivity and thresholds. **J. Hered.** 154: 143-151, 1963.

SENER RN et al. Bilateral split hand with bilateral tibial aplasia. **Pediatr. Radiol.** 19: 258-260, 1989.

SERTIÉ AL et al. Linkage analysis in a large brazilian family with van der Woude syndrome suggests the existence of a susceptibility locus for cleft palate at 17p11.2-11.1. **Am. J. Hum. Genet.** 65: 433-440, 1999.

SHENOY R, NUTAN K. Bilateral congenital split hand with tibial aplasia. **Indian J. Pediatr.** 71: 948-949, 2004.

SINGH SM, MURPHY B, O'REILLY R. Epigenetic contributors to the discordance of monozygotic twins. **Clin. Genet.** 62: 97-103, 2002.

SNYDER LH. Studies in human inheritance. IX. The inheritance of taste deficiency in man. **Ohio J. Sci.** 32: 436-440, 1932.

SNYDER LH. Studies in human inheritance. X. A table to determine the proportion of recessives to be expected in various matings involving a unit character. **Genetics** 19: 1-17, 1934.

SPRANGER M, SCHAPER J. Anomalous inheritance in a kindred with split hand, split foot malformation. **Eur. J. Pediatr.** **147**: 202-205, 1988.

STERN C. **Principles of human genetics**. 3.ed. San Francisco: WH Freeman, 1973.

SUAREZ BK, REICH T, TROST J. Limits of the general two-allele single locus model with incomplete penetrance. **Ann. Hum. Genet.** **40**: 231-244, 1976.

SUAREZ BK, FISHMAN PM, REICH T. Estimating the parameters of the incompletely penetrant single locus model using multiple populations. **Hum. Hered.** **27**: 336-351, 1977.

TAYLOR GL, PRIOR, AM. Blood groups in England. III. Discussion of the family material. **Ann. Eugen.** **9**: 18-44, 1939.

TRANKELL A. Aspects of genetics in psychology. **Am. J. Hum. Genet.** **7**: 264-276, 1955.

TRANKELL A. Penetrance calculus in population genetics. **Am. J. Hum. Genet.** **8**: 44-48, 1956.

VOGT O. Psychiatrisch wichtige Tatsachen der zoologisch-botanischen Systematik. **Z. Ges. Neurol. Psychiatr.** **101**: 805-832, 1926.

WILMOT GR, WARREN ST. A new mutational basis for disease. In: WELLS RD, WARREN ST. **Genetic instabilities and hereditary neurological diseases**. San Diego, California: Academic Press, 1998. p.3-12.

WILSON SR. Fitting of models of incomplete penetrance to family data. **Ann. Hum. Genet.** **35**: 99-108, 1971.

WITTERS I et al. Bilateral tibial agenesis with ectrodactyly (OMIM 119100): further evidence for autosomal recessive inheritance. **Am. J. Med. Genet.** 104: 209-213, 2001.

ZLOTOGORA J. On the inheritance of the split hand/split foot malformation. **Am. J. Med. Genet.** 53: 29-32, 1994.

ZLOTOGORA J. Penetrance and expressivity in the molecular age. **Genet. Med.** 5(5): 347-352, 2003.

ZLOTOGORA J, NUBANI N. Is there an autosomal recessive form of the split hand and split foot malformation? **J. Med. Genet.** 26: 138-140, 1989.

IX) ANEXO I

PROGRAMAS EM LINGUAGEM BASIC DESENVOLVIDOS UTILIZANDO O MICROSOFT® QBASIC®

TONGMOD1.BAS

```
REM MODEL # 1 : PEN(Rr) = K , PEN(RR) = 1-(1-K)^2
DEFDBL A-Z: DEFINT I-J
DIM A(2, 2), B(2, 2), C(2), CP(2), CK(2), C0(2)
CLS : LOCATE 10
INPUT "SAMPLE DESCRIPTION = "; A$
INPUT "N(DOM x DOM CROS.) = "; N1
INPUT "N(DOM x REC CROS.) = "; N2
INPUT "N(REC x REC CROS.) = "; N3
T0 = N1 + N2 + N3
INPUT "N(DOM | DOM x DOM) = "; N4
INPUT "N(REC | DOM x DOM) = "; N5
T1 = N4 + N5
INPUT "N(DOM | DOM x REC) = "; N6
INPUT "N(REC | DOM x REC) = "; N7
T2 = N6 + N7
INPUT "N(DOM | REC x REC) = "; N8
INPUT "N(REC | REC x REC) = "; N9
T3 = N8 + N9
T4 = N4 + N6 + N8: T5 = N5 + N7 + N9: T = T4 + T5
INPUT "INITIAL GUESSES (P,K) = "; P, K
PRINT "      MODEL I : PEN(RR) = 1-(1-K)^2 , PEN(Rr) = K": PRINT
PRINT "      "; A$: PRINT
PRINT "      p          s.e.(p)      K          s.e.(K)          dL/dp          dL/dK"
PRINT "      -----"
280 REM LOOP BEGINNING
GOSUB 720: C0(1) = C(1): C0(2) = C(2)
DP = P * .0000001: DK = K * .0000001: P = P + DP
GOSUB 720: CP(1) = C(1): CP(2) = C(2): P = P - DP: K = K + DK
GOSUB 720: CK(1) = C(1): CK(2) = C(2): K = K - DK
A(1, 1) = (C0(1) - CP(1)) / DP
A(1, 2) = ((C0(2) - CP(2)) / DP + (C0(1) - CK(1)) / DK) / 2
A(2, 1) = A(1, 2): A(2, 2) = (C0(2) - CK(2)) / DK
B(1, 1) = A(2, 2): B(1, 2) = -A(1, 2): B(2, 1) = -A(2, 1): B(2, 2) = A(1, 1)
FOR I = 1 TO 2: FOR J = 1 TO 2
B(I, J) = B(I, J) / (A(1, 1) * A(2, 2) - A(1, 2) * A(2, 1)): NEXT J: NEXT I
PRINT "      ";
PRINT USING "%.##### " ; P; SQR(B(1, 1)); K; SQR(B(2, 2));
PRINT USING "#####.##### " ; C(1); C(2)
P1 = P + B(1, 1) * C0(1) + B(1, 2) * C0(2)
K1 = K + B(2, 1) * C0(1) + B(2, 2) * C0(2)
IF ABS(P1 - P) < 1E-12 AND ABS(K1 - K) < 1E-12 THEN 710
P = P1: K = K1: GOTO 280
710 P = P1: K = K1: PRINT "      ";
PRINT "-----"
PRINT
P1 = ((2 - K) * (1 - P * K) / (2 - P * K)) ^ 2: Q1 = 1 - P1
P2 = (2 - K) * (1 - P * K * (2 - K)) / (2 - P * K): Q2 = 1 - P2
Q3 = P * K * (1 - K) * (2 - P * K * (3 - K)) / (1 - P * K) ^ 2: P3 = 1 - Q3
CS1 = N4 ^ 2 / (T1 * Q1) + N5 ^ 2 / (T1 * P1) - T1
CS2 = N6 ^ 2 / (T2 * Q2) + N7 ^ 2 / (T2 * P2) - T2
CS3 = N8 ^ 2 / (T3 * Q3) + N9 ^ 2 / (T3 * P3) - T3
CS4 = CS1 + CS2 + CS3
CS5 = (T4 - (T1 * Q1 + T2 * Q2 + T3 * Q3)) ^ 2 / (T1 * Q1 + T2 * Q2 + T3 * Q3)
CS5 = CS5 + (T5 - (T1 * P1 + T2 * P2 + T3 * P3)) ^ 2 / (T1 * P1 + T2 * P2 + T3 * P3)
CS6 = CS4 - CS5
PRINT "      RESULTS OF GOODNESS-OF-FIT TESTS : ": PRINT
PRINT USING "      a) off. of DxD parents : chi-square (1 d.f.) = ###.###"; CS1
PRINT USING "      b) off. of DxR parents : chi-square (1 d.f.) = ###.###"; CS2
PRINT USING "      c) off. of RxR parents : chi-square (1 d.f.) = ###.###"; CS3
PRINT USING "      d) 'total' chi-square : chi-square (3 d.f.) = ###.###"; CS4
PRINT USING "      e) 'pooled' chi-square : chi-square (1 d.f.) = ###.###"; CS5
PRINT USING "      f) heterogeneity c.s. : chi-square (2 d.f.) = ###.###"; CS6
END
720 REM SUBROUTINE FOR CALCULATING dL/dp [C(1)] AND dL/dK [C(2)]
```

```

N10 = 2 * N1 + N2 + N8
N11 = 2 * N1 + N2 - 2 * N4 - 2 * N5 - N6 - N7
N12 = 2 * N2 + 4 * N3 + 2 * N5 - 2 * N8 - 2 * N9
N13 = 2 * N5 + N7
PK1 = (2 - P * K) ^ 2 - (2 - K) ^ 2 * (1 - P * K) ^ 2
PK2 = 2 - P * K - (2 - K) * (1 - P * K * (2 - K))
PK3 = (1 - P * K) ^ 2 - P * K * (1 - K) * (2 - P * K * (3 - K))
C(1) = N10 / P - N11 * K / (2 - P * K) - N12 * K / (1 - P * K)
C(1) = C(1) + 2 * N4 * K * ((2 - K) ^ 2 * (1 - P * K) - (2 - P * K)) / PK1
C(1) = C(1) - N6 * K * (1 - (2 - K) ^ 2) / PK2
C(1) = C(1) - N7 * K * (2 - K) / (1 - P * K * (2 - K)) - N8 * K * (3 - K) / (2 - P *
K * (3 - K))
C(1) = C(1) - 2 * N9 * K * (1 - P * K + (1 - K) * (1 - P * K * (3 - K))) / PK3
C(2) = N10 / K - P * N11 / (2 - P * K) - P * N12 / (1 - P * K)
C(2) = C(2) - 2 * N4 * (P * (2 - P * K) - (2 - K) * (1 - P * K) * (1 + 2 * P * (1 -
K))) / PK1
C(2) = C(2) - N13 / (2 - K) + N6 * (1 - P + P * (2 - K) * (2 - 3 * K)) / PK2
C(2) = C(2) - 2 * N7 * P * (1 - K) / (1 - P * K * (2 - K)) - N8 / (1 - K)
C(2) = C(2) - N8 * P * (3 - 2 * K) / (2 - P * K * (3 - K))
C(2) = C(2) - N9 * P * (2 * (1 - P * K) + (1 - 2 * K) * (2 - P * K * (3 - K)) - P * K
* (1 - K) * (3 - 2 * K)) / PK3
RETURN

```

TONGTWI1.BAS

```

REM MODEL # 1 : PEN(Rr) = K , PEN(RR) = 1-(1-K)^2
DEFDBL A-Z: DEFINT I-J: O = 1: T = 2: R = 3: F = 4: M = .000000001#
DIM A(2, 2), B(2, 2), C(2), CP(2), CK(2), C0(2)
CLS : LOCATE 10
INPUT "SAMPLE DESCRIPTION = "; A$
INPUT "N(DOM - DOM MZ TWIN PAIRS) = "; N1
INPUT "N(DOM - REC MZ TWIN PAIRS) = "; N2
INPUT "N(REC - REC MZ TWIN PAIRS) = "; N3
N = N1 + N2 + N3: INPUT "INITIAL GUESSES (P,K) = "; P, K
PRINT USING "          N(DOM - DOM MZ TWIN PAIRS) = ####"; N1
PRINT USING "          N(DOM - REC MZ TWIN PAIRS) = ####"; N2
PRINT USING "          N(REC - REC MZ TWIN PAIRS) = ####"; N3: PRINT
PRINT "      MODEL 1 : PEN(RR) = 1-(1-K)^2 , PEN(Rr) = K": PRINT
PRINT "      "; : PRINT A$: PRINT
PRINT "          p          s.e.(p)      K          s.e.(K)          dL/dp          dL/dK"
PRINT "      -----"
280 REM LOOP BEGINNING
GOSUB 720: C0(1) = C(1): C0(2) = C(2)
DP = P * M: DK = K * M: P = P + DP
GOSUB 720: CP(1) = C(1): CP(2) = C(2): P = P - DP: K = K + DK
GOSUB 720: CK(1) = C(1): CK(2) = C(2): K = K - DK
A(1, 1) = (C0(1) - CP(1)) / DP
A(1, 2) = ((C0(2) - CP(2)) / DP + (C0(1) - CK(1)) / DK) / 2
A(2, 1) = A(1, 2): A(2, 2) = (C0(2) - CK(2)) / DK
B(1, 1) = A(2, 2): B(1, 2) = -A(1, 2): B(2, 1) = -A(2, 1): B(2, 2) = A(1, 1)
FOR I = 1 TO 2: FOR J = 1 TO 2
  B(I, J) = B(I, J) / (A(1, 1) * A(2, 2) - A(1, 2) * A(2, 1)): NEXT J: NEXT I
PRINT "      ";
PRINT USING "#.##### "; P; SQR(B(1, 1)); K; SQR(B(2, 2));
PRINT USING "#####.##### "; C(1); C(2)
P1 = P + B(1, 1) * C0(1) + B(1, 2) * C0(2)
K1 = K + B(2, 1) * C0(1) + B(2, 2) * C0(2)
IF ABS(P1 - P) < 1E-15 AND ABS(K1 - K) < 1E-15 THEN 710
P = P1: K = K1: GOTO 280
710 P = P1: K = K1: PRINT "      ";
PRINT "-----"
PRINT
KEX = 2 - (1 - SQR(N3 / N)) / (1 - SQR((N2 + 2 * N3) / (2 * N)))
PEX = (1 - SQR((N2 + 2 * N3) / (2 * N))) / K
PRINT "      EXPLICIT SOLUTIONS : ";
PRINT USING "p = #.#####"; PEX
PRINT "      ";
PRINT USING "K = #.#####"; KEX
END
720 REM SUBROUTINE FOR CALCULATING dL/dp [C(1)] AND dL/dK [C(2)]
PK1 = T * (O + P) - P * K * (F - K)
PK2 = T - P * K * (R - K): PK3 = O - P * K * (T - K)

```

```

C(1) = (N1 + N2) / P
C(1) = C(1) + N1 * (T - K * (F - K)) / PK1 - N2 * K * (R - K) / PK2
C(1) = C(1) - T * N3 * K * (T - K) / PK3
C(2) = (T * N1 + N2) / K
C(2) = C(2) - N2 / (O - K) - T * N1 * P * (T - K) / PK1
C(2) = C(2) - N2 * P * (R - T * K) / PK2 - F * N3 * P * (O - K) / PK3
RETURN

```

CONFINT3.BAS

```

DEFDBL A-Z: CLS
k0 = 0: kn = 1: GOSUB simpsonsrule: totalarea = area: k0 = .001: niter = 50
repeat:
K = k0: GOSUB functsub: pk0 = pk: K1 = 1
FOR j = 1 TO niter
  K1 = K1 - 1 / niter: K = K1: GOSUB functsub
  pk1 = pk: IF pk1 >= pk0 THEN GOTO getout
NEXT j
getout: k0 = k0: kn = K1: GOSUB simpsonsrule
IF area / totalarea >= .95 THEN k0 = k0 + 1 / niter: GOTO repeat
k0 = k0 - 1 / niter: K1 = K1 + 1 / niter: niter = niter * 5
IF ABS(.95 - area / totalarea) <= .00001 THEN END ELSE GOTO repeat
DO: LOOP WHILE INKEY$ <> " "
END
simpsonsrule:
h = (kn - k0) / 1000: K = k0: GOSUB functsub: s = pk
K = kn: GOSUB functsub: s = s + pk: w = 4
FOR j = 1 TO 999
  K = k0 + j * h
  GOSUB functsub
  s = s + w * pk
  w = 6 - w
NEXT j
area = h * s / 3: s = 0
IF k0 <> 0 THEN
  'LOCATE 10
  PRINT USING "#.##### #.##### #.##### "; k0; kn; area / totalarea
  K = k0: GOSUB functsub: pk0 = pk
  K = kn: GOSUB functsub: pk1 = pk
  'PRINT USING "####.##### "; pk0; pk1
END IF
RETURN
functsub:
pk = K ^ 12 * (1 - K) ^ 11 * (2 - K) ^ 14
pk = pk * (3 - 3 * K + K ^ 2)
pk = pk * (1 / 2 + (1 - K) / 2 * (2 - K) / 2) ^ 2
pk = pk * (1 / 2 + (1 - K) / 2 * ((2 - K) / 2) ^ 2) ^ 4
pk = pk * (1 / 2 + (1 - K) / 2 * ((2 - K) / 2) ^ 3) ^ 3
RETURN

```

KVAR_4.BAS

```

CLS : DEFDBL A-Z: K1 = .99: K2 = .99: K3 = .01: K4 = .01
IF K1 >= 1 THEN K1 = .99
IF K1 <= 0 THEN K1 = .01
IF K2 >= 1 THEN K2 = .99
IF K2 <= 0 THEN K2 = .01
IF K3 >= 1 THEN K3 = .99
IF K3 <= 0 THEN K3 = .01

```

```

IF K4 >= 1 THEN K4 = .99
IF K4 <= 0 THEN K4 = .01
CLS : EPS = .000000001#
PRINT " i      K1      K2      K3      K4      dL/dK1      dL/dK2 dL/dK3
PRINT "-----
LOOPAGAIN:
'DK1 = dL/dK1
'DK2 = dL/dK2
'DK3 = dL/dK3
'DK4 = dL/dK4
'A(i,j) : matrix of second derivatives
'A(1,1) = d^2L/dK1^2
'A(1,2) = A(2,1) = d^2L/dK1K2
'A(1,3) = A(3,1) = d^2L/dK1K3
'A(1,4) = A(4,1) = d^2L/dK1K4
'A(2,2) = d^2L/dK2^2
'A(2,3) = A(3,2) = d^2L/dK2K3
'A(2,4) = A(4,2) = d^2L/dK2K4
'A(3,3) = d^2L/dK3^2
'A(3,4) = A(4,3) = d^2L/dK3K4
'A(4,4) = d^2L/dK4^2
'B(i,j) = -A(i,j)^(-1) : variance-covariance matrix
'B(1,1) = var(K1)
'B(1,2) = B(2,1) = cov(K1,K2)
'B(1,3) = B(3,1) = cov(K1,K3)
'B(1,4) = B(4,1) = cov(K1,K4)
'B(2,2) = var(K2)
'B(2,3) = B(3,2) = cov(K2,K3)
'B(2,4) = B(4,2) = cov(K2,K4)
'B(3,3) = var(K3)
'B(3,4) = B(4,3) = cov(K3,K4)
'B(4,4) = var(K4)
DK1 = 1/K1 - 5/(1 - K1) - 2/(2 - K1)
DK1 = DK1 - (2-K2)^2/(4+(1-K1)*(2-K2)^2)
DK2 = 1/K2 - 5/(1-K2) - (2-K3)/(2+(1-K2)*(2-K3))
DK2 = DK2 - 2*(1-K1)*(2-K2)/(4+(1-K1)*(2-K2)^2)
DK2 = DK2 - 2*(2-K3)^3/(8+(1-K2)*(2-K3)^3)
DK3 = 7/K3 - 1/(1-K3) - 4/(2-K3)
DK3 = DK3 - (1-K2)/(2+(1-K2)*(2-K3))
DK3 = DK3 - (2-K4)/(2+(1-K3)*(2-K4))
DK3 = DK3 - 3*(2-K4)^2/(4+(1-K3)*(2-K4)^2)
DK3 = DK3 - 6*(1-K2)*(2-K3)^2/(8+(1-K2)*(2-K3)^3)
DK3 = DK3 - (2-K4)^3/(8+(1-K3)*(2-K4)^3)
DK3 = DK3 - (3-2*K3)/(3-3*K3+K3^2)
DK4 = 3/K4 - 8/(2-K4) - (1-K3)/(2+(1-K3)*(2-K4))
DK4 = DK4 - 6*(1-K3)*(2-K4)/(4+(1-K3)*(2-K4)^2)
DK4 = DK4 - 3*(1-K3)*(2-K4)^2/(8+(1-K3)*(2-K4)^3)
A(1,1) = -5/(1-K1)^2 - 2/(2-K1)^2 - 1/K1^2
A(1,1) = A(1,1) - (2-K2)^4/(4+(1-K1)*(2-K2)^2)^2
A(1,2) = 2*(2-K2)/(4+(1-K1)*(2-K2)^2)
A(1,2) = A(1,2) - 2*(1-K1)*(2-K2)^3/(4+(1-K1)*(2-K2)^2)^2
A(1,3) = 0: A(1,4) = 0: A(2,1) = A(1,2)
A(2,2) = 2*(1-K1)/(4+(1-K1)*(2-K2)^2)
A(2,2) = A(2,2) - 5/(1-K2)^2 - 1/K2^2
A(2,2) = A(2,2) - 4*(1-K1)^2*(2-K2)^2/(4+(1-K1)*(2-K2)^2)^2
A(2,2) = A(2,2) - 2*(2-K3)^6/(8+(1-K2)*(2-K3)^3)^2
A(2,2) = A(2,2) + (2-K3)*(-2+K3)/(2+(1-K2)*(2-K3))^2
A(2,3) = 1/(2+(1-K2)*(2-K3)) - (1-K2)*(2-K3)/(2+(1-K2)*(2-K3))^2
A(2,3) = A(2,3) + 6*(2-K3)^2/(8+(1-K2)*(2-K3)^3)
A(2,3) = A(2,3) - 6*(1-K2)*(2-K3)^5/(8+(1-K2)*(2-K3)^3)^2
A(2,4) = 0: A(3,1) = A(1,3): A(3,2) = A(2,3)
A(3,3) = (1-K2)*(-1+K2)/(2+(1-K2)*(2-K3))^2 - 1/(1-K3)^2
A(3,3) = A(3,3) - 4/(2-K3)^2
A(3,3) = A(3,3) + 12*(1-K2)*(2-K3)/(8+(1-K2)*(2-K3)^3)
A(3,3) = A(3,3) - 18*(1-K2)^2*(2-K3)^4/(8+(1-K2)*(2-K3)^3)
A(3,3) = A(3,3) - 7/K3^2
A(3,3) = A(3,3) - 3*(2-K4)^4/(4+(1-K3)*(2-K4)^2)^2

```

```

A(3,3) = A(3,3) - (2-K4)^6/(8+(1-K3)*(2-K4)^3)^2
A(3,3) = A(3,3) - (2-K4)*(-2+K4)/(2+(1-K3)*(2-K4))^2
A(3,3) = A(3,3) + (3-2*K3)*(-3+2*K3)/(3-3*K3+K3^2)^2
A(3,3) = A(3,3) + 2/(3-3*K3+K3^2)
A(3,4) = 1/(2+(1-K3)*(2-K4)) - (1-K3)*(2-K4)/(2+(1-K3)*(2-K4))^2
A(3,4) = A(3,4) + 6*(2-K4)/(4+(1-K3)*(2-K4))^2
A(3,4) = A(3,4) + 3*(2-K4)^2/(8+(1-K3)*(2-K4)^3)
A(3,4) = A(3,4) - 6*(1-K3)*(2-K4)^3/(4+(1-K3)*(2-K4)^2)^2
A(3,4) = A(3,4) - 3*(1-K3)*(2-K4)^5/(8+(1-K3)*(2-K4)^3)^2
A(4,1) = A(1,4): A(4,2) = A(2,4): A(4,3) = A(3,4)
A(4,4) = (1-K3)*(-1+K3)/(2+(1-K3)*(2-K4))^2
A(4,4) = A(4,4) + 6*(1-K3)/(4+(1-K3)*(2-K4)^2)
A(4,4) = A(4,4) - 8/(2-K4)^2
A(4,4) = A(4,4) + 6*(1-K3)*(2-K4)/(8+(1-K3)*(2-K4)^3)
A(4,4) = A(4,4) - 12*(1-K3)^2*(2-K4)^2/(4+(1-K3)*(2-K4)^2)^2
A(4,4) = A(4,4) - 9*(1-K3)^2*(2-K4)^4/(8+(1-K3)*(2-K4)^3)^2 - 3/K4^2
FOR I = 1 TO 4: FOR J = 1 TO 4: A(I, J) = -A(I, J): NEXT J: NEXT I
'here begins the matrix inversion procedure
FOR J = 1 TO 4: B(J, J) = 1: NEXT J
FOR J = 1 TO 4: I = J - 1
LOOP1: I = I + 1
IF A(I, J) <> 0 THEN GOTO LOOP2
GOTO LOOP1
LOOP2: FOR K = 1 TO 4: SWAP A(J, K), A(I, K): SWAP B(J, K), B(I, K): NEXT K
TT = 1 / A(J, J)
FOR K = 1 TO 4: A(J, K) = TT * A(J, K): B(J, K) = TT * B(J, K): NEXT K
FOR L = 1 TO 4: IF L = J THEN GOTO NEXTL
TT = -A(L, J)
FOR K = 1 TO 4
A(L, K) = A(L, K) + TT * A(J, K): B(L, K) = B(L, K) + TT * B(J, K)
NEXT K
NEXTL: NEXT L
NEXT J
'here ends the matrix inversion procedure
K11 = K1 + DK1 * B(1,1) + DK2 * B(1,2) + DK3 * B(1,3) + DK4 * B(1,4)
K21 = K2 + DK1 * B(2,1) + DK2 * B(2,2) + DK3 * B(2,3) + DK4 * B(2,4)
K31 = K3 + DK1 * B(3,1) + DK2 * B(3,2) + DK3 * B(3,3) + DK4 * B(3,4)
K41 = K4 + DK1 * B(4,1) + DK2 * B(4,2) + DK3 * B(4,3) + DK4 * B(4,4)
PRINT USING " ## "; ITERATION;
PRINT USING "#.##### "; K1; K2; K3; K4;
PRINT USING "##### "; DK1; DK2; DK3; DK4
DO: LOOP WHILE INKEY$ <> " "
IF ABS(K11-K1) > EPS AND ABS(K21-K2) > EPS AND ABS(K31-K3) > EPS AND ABS(K41-K4) > EPS
THEN
    K1 = K11: K2 = K21: K3 = K31: K4 = K41
    FOR I = 1 TO 4: FOR J = 1 TO 4: B(I, J) = 0: NEXT J, I
    ITERATION = ITERATION + 1: GOTO LOOPAGAIN
ELSE
PRINT IF ABS(K11 - K1) > EPS AND ABS(K21 - K2) > EPS AND ABS(K31 - K3) > EPS AND ABS
    K1 = K11: K2 = K21: K3 = K31: K4 = K41
    FOR I = 1 TO 4: FOR J = 1 TO 4: B(I, J) = 0: NEXT J, I
    ITERATION = ITERATION + 1: GOTO LOOPAGAIN
ELSE
IF ABS(K11 - K1) > EPS AND ABS(K21 - K2) > EPS AND ABS(K31 - K3) > EPS AND ABS
    K1 = K11: K2 = K21: K3 = K31: K4 = K41
    FOR I = 1 TO 4: FOR J = 1 TO 4: B(I, J) = 0: NEXT J, I
    ITERATION = ITERATION + 1: GOTO LOOPAGAIN
ELSE
PRINT "-----IF ABS(K11 - K1) > EPS
AND ABS(K21 - K2) > EPS AND ABS(K31 - K3) > EPS AND ABS
    K1 = K11: K2 = K21: K3 = K31: K4 = K41
    FOR I = 1 TO 4: FOR J = 1 TO 4: B(I, J) = 0: NEXT J, I
    ITERATION = ITERATION + 1: GOTO LOOPAGAIN
ELSE
    PRINT "-----
END IF
'B(1, 2) = (B(1, 2) + B(2, 1)) / 2: B(2, 1) = B(1, 2)

```



```

'B(1, 3) = (B(1, 3) + B(3, 1)) / 2: B(3, 1) = B(1, 3)
'B(1, 4) = (B(1, 4) + B(4, 1)) / 2: B(4, 1) = B(1, 4)
'B(2, 3) = (B(2, 3) + B(3, 2)) / 2: B(3, 2) = B(2, 3)
'B(2, 4) = (B(2, 4) + B(4, 2)) / 2: B(4, 2) = B(2, 4)
'B(3, 4) = (B(3, 4) + B(4, 3)) / 2: B(4, 3) = B(3, 4)
PRINT : PRINT "VARIANCE-COVARIANCE MATRIX var(i,j), i, j = 1,...,4"
PRINT
PRINT USING " ##### "; 1; 2; 3; 4
FOR I = 1 TO 4: PRINT USING "# "; I;
FOR J = 1 TO 4
PRINT USING " ##### "; B(I, J);
IF J / 4 = INT(J / 4) THEN PRINT
NEXT J, I
PRINT : PRINT "APPR. 95% CONFID. INTERVALS OF Ki ESTIMATES"
PRINT : PRINT "i      Ki      ll(Ki) - ul(Ki)"
PRINT "-----"
K(1) = K1: K(2) = K2: K(3) = K3: K(4) = K4
FOR I = 1 TO 4
LLK(I) = K(I) - 1.96 * SQR(B(I, I)): ULK(I) = K(I) + 1.96 * SQR(B(I, I))
IF LLK(I) < 0 THEN LLK(I) = 0
IF ULK(I) > 1 THEN ULK(I) = 1
PRINT USING "#      #####      #.#### - #.####"; I; K(I); LLK(I); ULK(I)
NEXT I
PRINT "-----"
END IF
'B(1, 2) = (B(1, 2) + B(2, 1)) / 2: B(2, 1) = B(1, 2)
'B(1, 3) = (B(1, 3) + B(3, 1)) / 2: B(3, 1) = B(1, 3)
'B(1, 4) = (B(1, 4) + B(4, 1)) / 2: B(4, 1) = B(1, 4)
'B(2, 3) = (B(2, 3) + B(3, 2)) / 2: B(3, 2) = B(2, 3)
'B(2, 4) = (B(2, 4) + B(4, 2)) / 2: B(4, 2) = B(2, 4)
'B(3, 4) = (B(3, 4) + B(4, 3)) / 2: B(4, 3) = B(3, 4)
PRINT : PRINT "VARIANCE-COVARIANCE MATRIX var(i,j), i, j = 1,...,4"
PRINT
PRINT USING " ##### "; 1; 2; 3; 4
FOR I = 1 TO 4: PRINT USING "# "; I;
FOR J = 1 TO 4
PRINT USING " ##### "; B(I, J);
IF J / 4 = INT(J / 4) THEN PRINT
NEXT J, I
PRINT : PRINT "APPR. 95% CONFID. INTERVALS OF Ki ESTIMATES"
PRINT : PRINT "i      Ki      ll(Ki) - ul(Ki)"
PRINT "-----"
K(1) = K1: K(2) = K2: K(3) = K3: K(4) = K4
FOR I = 1 TO 4
LLK(I) = K(I) - 1.96 * SQR(B(I, I)): ULK(I) = K(I) + 1.96 * SQR(B(I, I))
IF LLK(I) < 0 THEN LLK(I) = 0
IF ULK(I) > 1 THEN ULK(I) = 1
PRINT USING "#      #####      #.#### - #.####"; I; K(I); LLK(I); ULK(I)
NEXT I
PRINT "-----"
END IF
'B(1, 2) = (B(1, 2) + B(2, 1)) / 2: B(2, 1) = B(1, 2)
'B(1, 3) = (B(1, 3) + B(3, 1)) / 2: B(3, 1) = B(1, 3)
'B(1, 4) = (B(1, 4) + B(4, 1)) / 2: B(4, 1) = B(1, 4)
'B(2, 3) = (B(2, 3) + B(3, 2)) / 2: B(3, 2) = B(2, 3)
'B(2, 4) = (B(2, 4) + B(4, 2)) / 2: B(4, 2) = B(2, 4)
'B(3, 4) = (B(3, 4) + B(4, 3)) / 2: B(4, 3) = B(3, 4)
PRINT : PRINT "VARIANCE-COVARIANCE MATRIX var(i,j), i, j = 1,...,4"
PRINT
PRINT USING " ##### "; 1; 2; 3; 4
FOR I = 1 TO 4: PRINT USING "# "; I;
FOR J = 1 TO 4
PRINT USING " ##### "; B(I, J);
IF J / 4 = INT(J / 4) THEN PRINT
NEXT J, I
PRINT : PRINT "APPR. 95% CONFID. INTERVALS OF Ki ESTIMATES"
PRINT : PRINT "i      Ki      ll(Ki) - ul(Ki)"

```

```

        PRINT "-----"
K(1) = K1: K(2) = K2: K(3) = K3: K(4) = K4
FOR I = 1 TO 4
LLK(I) = K(I) - 1.96 * SQR(B(I, I)): ULK(I) = K(I) + 1.96 * SQR(B(I, I))
IF LLK(I) < 0 THEN LLK(I) = 0
IF ULK(I) > 1 THEN ULK(I) = 1
PRINT USING "#      #.#####      #.#### - #.####"; I; K(I); LLK(I); ULK(I)
NEXT I
PRINT "-----"
IF ABS(K11 - K1) > EPS AND ABS(K21 - K2) > EPS AND ABS(K31 - K3) > EPS AND ABS
    K1 = K11: K2 = K21: K3 = K31: K4 = K41
    FOR I = 1 TO 4: FOR J = 1 TO 4: B(I, J) = 0: NEXT J, I
    ITERATION = ITERATION + 1: GOTO LOOPAGAIN
ELSE
PRINT "-----"
END IF
'B(1, 2) = (B(1, 2) + B(2, 1)) / 2: B(2, 1) = B(1, 2)
'B(1, 3) = (B(1, 3) + B(3, 1)) / 2: B(3, 1) = B(1, 3)
'B(1, 4) = (B(1, 4) + B(4, 1)) / 2: B(4, 1) = B(1, 4)
'B(2, 3) = (B(2, 3) + B(3, 2)) / 2: B(3, 2) = B(2, 3)
'B(2, 4) = (B(2, 4) + B(4, 2)) / 2: B(4, 2) = B(2, 4)
'B(3, 4) = (B(3, 4) + B(4, 3)) / 2: B(4, 3) = B(3, 4)
PRINT : PRINT "VARIANCE-COVARIANCE MATRIX var(i,j), i, j = 1,...,4"
PRINT
PRINT USING " ##### " ; 1; 2; 3; 4
FOR I = 1 TO 4: PRINT USING "# "; I;
FOR J = 1 TO 4
PRINT USING " ##### " ; B(I, J);
IF J / 4 = INT(J / 4) THEN PRINT
NEXT J, I
PRINT : PRINT "APPR. 95% CONFID. INTERVALS OF Ki ESTIMATES"
PRINT : PRINT "i      Ki      ll(Ki) - ul(Ki)"
PRINT "-----"
K(1) = K1: K(2) = K2: K(3) = K3: K(4) = K4
FOR I = 1 TO 4
LLK(I) = K(I) - 1.96 * SQR(B(I, I)): ULK(I) = K(I) + 1.96 * SQR(B(I, I))
IF LLK(I) < 0 THEN LLK(I) = 0
IF ULK(I) > 1 THEN ULK(I) = 1
PRINT USING "#      #.#####      #.#### - #.####"; I; K(I); LLK(I); ULK(I)
NEXT I
PRINT "-----"
END IF
'B(1, 2) = (B(1, 2) + B(2, 1)) / 2: B(2, 1) = B(1, 2)
'B(1, 3) = (B(1, 3) + B(3, 1)) / 2: B(3, 1) = B(1, 3)
'B(1, 4) = (B(1, 4) + B(4, 1)) / 2: B(4, 1) = B(1, 4)
'B(2, 3) = (B(2, 3) + B(3, 2)) / 2: B(3, 2) = B(2, 3)
'B(2, 4) = (B(2, 4) + B(4, 2)) / 2: B(4, 2) = B(2, 4)
'B(3, 4) = (B(3, 4) + B(4, 3)) / 2: B(4, 3) = B(3, 4)
PRINT : PRINT "VARIANCE-COVARIANCE MATRIX var(i,j), i, j = 1,...,4"
PRINT
PRINT USING " ##### " ; 1; 2; 3; 4
FOR I = 1 TO 4: PRINT USING "# "; I;
FOR J = 1 TO 4
PRINT USING " ##### " ; B(I, J);
IF J / 4 = INT(J / 4) THEN PRINT
NEXT J, I
PRINT : PRINT "APPR. 95% CONFID. INTERVALS OF Ki ESTIMATES"
PRINT : PRINT "i      Ki      ll(Ki) - ul(Ki)"
PRINT "-----"
K(1) = K1: K(2) = K2: K(3) = K3: K(4) = K4
FOR I = 1 TO 4
LLK(I) = K(I) - 1.96 * SQR(B(I, I)): ULK(I) = K(I) + 1.96 * SQR(B(I, I))
IF LLK(I) < 0 THEN LLK(I) = 0
IF ULK(I) > 1 THEN ULK(I) = 1
PRINT USING "#      #.#####      #.#### - #.####"; I; K(I); LLK(I); ULK(I)
NEXT I
PRINT "-----"

```

KVAR_2.BAS

```

CLS : DEFDBL A-Z
K1 = .99: K2 = .01
CLS : EPS = .000000001#
PRINT " i   K1   K2   dL/dK1   dL/dK2   var(K1)   cov(K1,K2)   var(K1) "
PRINT "-----"
LOOPAGAIN:
'DK1 = dL/dK1
'DK2 = dL/dK2
'D11 = d^2L/dK1^2
'D12 = D21 = d^2L/dK1K2 = d^2L/dK2K1
'D22 = d^2L/dK2^2
'M11 = D22/(D11.D22-D12.D21) = var(K1)
'M12 = M21 = -D12/(D11.D22-D12.D21) = -D21/(D11.D22-D12.D21) = cov(K1,K2)
'M22 = D11/(D11.D22-D12.D21) = VAR(K2)
K3 = 1 - K1: K4 = 2 - K1: K5 = 1 - K2: K6 = 2 - K2
DK1 = 2/K1 - 10/K3 - 2/K4
DK1 = DK1 - K6/(2+K3*K6)
DK1 = DK1 + (-2*K3*K4-K4^2)/(4+K3*K4^2)
DK1 = DK1 - 2*K6^3/(8+K3*K6^3)
DK2 = 10/K2 - 1/K5 - 12/K6
DK2 = DK2 - K3/(2+K3*K6)
DK2 = DK2 + (-3+2*K2)/(2+K5*K6)
DK2 = DK2 + (3*(-2*K5*K6-K6^2))/(4+K5*K6^2)
DK2 = DK2 - 6*K3*K6^2/(8+K3*K6^3)
DK2 = DK2 + (-3*K5*K6^2-K6^3)/(8+K5*K6^3)
DK2 = DK2 + (-3+2*K2)/(3-3*K2+K2^2)
D11 = (2*K3+4*K4)/(4+K3*K4^2) - 10/K3^2
D11 = D11 - 2/K4^2 - 2/K1^2
D11 = D11 - K6^2/(2+K3*K6)^2
D11 = D11 - ((-2*K3*K4-K4^2)^2)/(4+K3*K4^2)^2
D11 = D11 - 2*K6^6/(8+K3*K6^3)^2
D12 = 1/(2+K3*K6)
D12 = D12 - K3*K6/(2+K3*K6)^2
D12 = D12 + 6*K6^2/(8+K3*K6^3)
D12 = D12 - 6*K3*K6^5/(8+K3*K6^3)^2
D22 = -K3^2/(2+K3*K6)^2
D22 = D22 + 2/(2+K5*K6)
D22 = D22 - 3*(-2*K5*K6-K6^2)^2/(4+K5*K6^2)^2
D22 = D22 - 1/K5^2 - 12/K6^2
D22 = D22 + (3*(2*K5+4*K6))/(4+K5*K6^2)
D22 = D22 + 12*K3*K6/(8+K3*K6^3)
D22 = D22 + (6*K5*K6+6*K6^2)/(8+K5*K6^3)
D22 = D22 - ((-3*K5*K6^2-K6^3)^2)/(8+K5*K6^3)^2
D22 = D22 - 18*K3^2*K6^4/(8+K3*K6^3)^2
D22 = D22 - 10/K2^2
D22 = D22 - (3-2*K2)^2/(2+K5*K6)^2
D22 = D22 - (3-2*K2)^2/(3-3*K2+K2^2)^2
D22 = D22 + 2/(3-3*K2+K2^2)
DET = D11 * D22 - D12 * D12
M11 = -D22 / DET: M12 = D12 / DET: M21 = M12: M22 = -D11 / DET
K11 = K1 + M11 * DK1 + M12 * DK2
K21 = K2 + M21 * DK1 + M22 * DK2
PRINT USING " ## "; ITERATION;
PRINT USING "#.##### "; K1; K2;
PRINT USING "##### "; DK1; DK2;
PRINT USING "###.##### "; M11; M12; M22
IF ABS(K11 - K1) > EPS AND ABS(K21 - K2) > EPS THEN
    K1 = K11: K2 = K21: ITERATION = ITERATION + 1: GOTO LOOPAGAIN
ELSE
PRINT "-----"
    END
END IF

```

MAJ15_KC.BAS

```
DEFDBL A-Z: CLS
OPTION BASE 0: DIM g(10)
OPEN "c:\temp\majewski.txt" FOR OUTPUT AS #1
' values of g(i) = e^(-2).2^i/i!, i = 0,...,9
g(0) = .13533528324#: g(1) = .27067056647#: g(2) = .27067056647#
g(3) = .18044704432#: g(4) = .09022352216#: g(5) = .03608940886#
g(6) = .01202980295#: g(7) = .00343708656#: g(8) = .00085927164#
g(9) = .00019094925#
FOR i = 0 TO 9: FOR j = 0 TO 9: FOR l = 0 TO 9: FOR m = 0 TO 9
    IF i + j + l + m = 15 THEN gsum = gsum + g(i) * g(j) * g(l) * g(m)
NEXT m, l, j, i
tthou = 100000: D1 = CDBL(1 / tthou): D2 = CDBL(2 / tthou)
PRINT #1, " i j l m k f(k) k.f(k) "
PRINT #1, " -----"
FOR i = 0 TO 9: FOR j = 0 TO 9: FOR l = 0 TO 9: FOR m = 0 TO 9
    k = .5
    IF i + j + l + m = 15 THEN
        LONGLOOP:
        k0 = CDBL(k): k1 = CDBL(k + k * D1): k2 = CDBL(k + k * D2)
        k = k0: GOSUB LK0FUNC: L0 = F ' L0 = L(k)
        k = k1: GOSUB LK0FUNC: L1 = F ' L1 = L(k+Dk)
        k = k2: GOSUB LK0FUNC: L2 = F ' L2 = L(k+2Dk)
        k = k0
        DLDK = CDBL((L1 - L0) / (k * D1))
        D2LDK2 = CDBL(((L2 - L1) / (k * D1) - DLDK) / (k * D1))
        KONE = CDBL(k - DLDK / D2LDK2)
        IF ABS(KONE - k) < 1E-08 THEN GOTO EXACT0 ELSE k = KONE: GOTO LONGLOOP
    EXACT0:
    PRINT #1, USING " ## ## ## ## ##.##### "; i; j; l; m; k;
    freqk = g(i) * g(j) * g(l) * g(m) / gsum
    PRINT #1, USING "##### "; freqk; k * freqk
    totk = totk + freqk
    ksum = ksum + k * freqk
    k2sum = k2sum + k * k * freqk
    END IF
NEXT m, l, j, i
PRINT #1, " -----"
PRINT #1, " ";
PRINT #1, USING "##### "; totk; ksum
kvar = k2sum - ksum * ksum: kse = SQR(kvar)
PRINT #1, USING "k = #####"; ksum
PRINT #1, USING "se(k) = #####"; kse
CLOSE #1
END
LK0FUNC:
F = 4 * LOG(k) + 2 * LOG(1 - k) + 5 * LOG(2 - k)
F = F + LOG(1 / 2 + (((1 - k) / 2) * ((2 - k) / 2) ^ 9))
F = F + LOG(1 / 2 + (((1 - k) / 2) * ((2 - k) / 2) ^ i))
F = F + LOG(1 / 2 + (((1 - k) / 2) * ((2 - k) / 2) ^ j))
F = F + LOG(1 / 2 + (((1 - k) / 2) * ((2 - k) / 2) ^ l))
F = F + LOG(1 / 2 + (((1 - k) / 2) * ((2 - k) / 2) ^ m))
RETURN
```

MAJ15_2K.BAS

```
DEFDBL A-Z: CLS
OPTION BASE 0: DIM g(16)
OPEN "c:\temp\maj4_2k.txt" FOR OUTPUT AS #1
' values of g(i) = e^(-2).2^i/i!, i = 0,...,9
g(0) = .13533528324#: g(1) = .27067056647#: g(2) = .27067056647#
```

```

g(3) = .18044704432#: g(4) = .09022352216#: g(5) = .03608940886#
g(6) = .01202980295#: g(7) = .00343708656#: g(8) = .00085927164#
g(9) = .00019094925#
FOR i = 0 TO 9: FOR j = 0 TO 9: FOR l = 0 TO 9: FOR m = 0 TO 9
  IF i + j + l + m = 15 THEN gsum = gsum + g(i) * g(j) * g(l) * g(m)
NEXT m, l, j, i
tthou = 100000: D1 = CDBL(1 / tthou): D2 = CDBL(2 / tthou)
FOR i = 0 TO 9: FOR j = 0 TO 9: FOR l = 0 TO 9: FOR m = 0 TO 9
  IF i + j + l + m = 15 THEN
    EPS = .000000001#: DELTA = .0001#
LOOPAGAIN:
  ' DK1 = dL/dK1
  ' DK2 = dL/dK2
  ' D11 = d^2L/dK1^2
  ' D12 = D21 = d^2L/dK1K2 = d^2L/dK2K1
  ' D22 = d^2L/dK2^2
  ' M11 = -D22/(D11.D22-D12.D21) = var(K1)
  ' M12 = M21 = D12/(D11.D22-D12.D21) = D21/(D11.D22-D12.D21) = cov(K1,K2)
  ' M22 = -D11/(D11.D22-D12.D21) = var(K2)
GOSUB DERIVDK: DK10 = DK1: DK20 = DK2: DELTAK1 = DELTA * K1
K1 = K1 + DELTAK1: GOSUB DERIVDK: DK11 = DK1: DK21 = DK2
DK1 = DK10: K1 = K1 - DELTAK1: DELTAK2 = DELTA * K2
K2 = K2 + DELTAK2: GOSUB DERIVDK: DK12 = DK1: DK22 = DK2
DK2 = DK20: K2 = K2 - DELTAK2
'DK10 = (dL/dK1)K1,K2
'DK11 = (dL/dK1)K1+DELTAK1,K2
'DK12 = (dL/dK1)K1,K2+DELTAK2
'DK20 = (dL/dK2)K1,K2
'DK21 = (dL/dK2)K1+DELTAK1,K2
'DK22 = (dL/dK2)K1,K2+DELTAK2
D11 = (DK11 - DK10) / DELTAK1
D22 = (DK22 - DK20) / DELTAK2
D12 = ((DK21 - DK20) / DELTAK1 + (DK12 - DK10) / DELTAK2) / 2
DET = D11 * D22 - D12 * D12
M11 = -D22 / DET: M12 = D12 / DET: M21 = M12: M22 = -D11 / DET
K11 = K1 + M11 * DK1 + M12 * DK2
K21 = K2 + M21 * DK1 + M22 * DK2
IF ABS(K11 - K1) > EPS AND ABS(K21 - K2) > EPS THEN
  K1 = K11: K2 = K21: ITERATION = ITERATION + 1: GOTO LOOPAGAIN
ELSE
  PRINT USING "## ## ## ## #.#### #.#### "; i; j; l; m; K1; K2;
  PRINT USING " #.##### #.##### "; freqK1; freqK2
  PRINT #1, USING "## ## ## ## #.#### #.#### "; i; j; l; m; K1; K2;
  PRINT #1, USING " #.##### #.##### "; freqK1; freqK2
  freqK1 = g(i) * g(j) * g(l) * g(m) / gsum
  freqK2 = g(i) * g(j) * g(l) * g(m) / gsum
  totK1 = totK1 + freqK1
  totK2 = totK2 + freqK2
  ksum1 = ksum1 + K1 * freqK1
  ksum2 = ksum2 + K2 * freqK2
  k2sum1 = k2sum1 + K1 * K1 * freqK1
  k2sum2 = k2sum2 + K2 * K2 * freqK2
END IF
END IF
NEXT m, l, j, i
PRINT
PRINT #1,
PRINT USING " #.#### #.#### "; ksum1; ksum2;
PRINT USING " #.##### #.##### "; totK1; totK2
PRINT #1, USING " #.#### #.#### "; ksum1; ksum2;
PRINT #1, USING " #.##### #.##### "; totK1; totK2
kvar1 = (k2sum1 - ksum1 * ksum1)
kvar2 = (k2sum2 - ksum2 * ksum2)
seK1 = SQR(kvar1)
seK2 = SQR(kvar2)
PRINT
PRINT #1,

```

```

PRINT USING "K1      = #####"; ksum1
PRINT USING "se(K1) = #####"; seK1
PRINT #1, USING "K1      = #####"; ksum1
PRINT #1, USING "se(K1) = #####"; seK1
PRINT
PRINT #1,
PRINT USING "K2      = #####"; ksum2
PRINT USING "se(K2) = #####"; seK2
PRINT #1, USING "K2      = #####"; ksum2
PRINT #1, USING "se(K2) = #####"; seK2
END
DERIVDK:
K3 = 1 - K1: K4 = 2 - K1: K5 = 1 - K2: K6 = 2 - K2
DK1 = -2 / K3 - 3 / K4 + 2 / K1 - K6 ^ 9 / (1024 * (1 / 2 + (K3 * K6 ^ 9) / 1024))
DK1 = DK1 - (1 / 2) ^ i * K6 ^ i / (2 * (1 / 2 + ((1 / 2) ^ i * K3 * K6 ^ i) / 2))
DK1 = DK1 - (1 / 2) ^ j * K6 ^ j / (2 * (1 / 2 + ((1 / 2) ^ j * K3 * K6 ^ j) / 2))
DK1 = DK1 - (1 / 2) ^ l * K6 ^ l / (2 * (1 / 2 + ((1 / 2) ^ l * K3 * K6 ^ l) / 2))
DK1 = DK1 - (1 / 2) ^ m * K6 ^ m / (2 * (1 / 2 + ((1 / 2) ^ m * K3 * K6 ^ m) / 2))
DK2 = -2 / K6 + 2 / K2 - 9 * K3 * K6 ^ 8 / (1024 * (1 / 2 + (K3 * K6 ^ 9) / 1024))
DK2 = DK2 - (1 / 2) ^ i * i * K3 * K6 ^ (i - 1) / (2 * (1 / 2 + ((1 / 2) ^ i * K3 * K6 ^ i) / 2))
DK2 = DK2 - (1 / 2) ^ j * j * K3 * K6 ^ (j - 1) / (2 * (1 / 2 + ((1 / 2) ^ j * K3 * K6 ^ j) / 2))
DK2 = DK2 - (1 / 2) ^ l * l * K3 * K6 ^ (l - 1) / (2 * (1 / 2 + ((1 / 2) ^ l * K3 * K6 ^ l) / 2))
DK2 = DK2 - (1 / 2) ^ m * m * K3 * K6 ^ (m - 1) / (2 * (1 / 2 + ((1 / 2) ^ m * K3 * K6 ^ m) / 2))

```

MAJ21_KC.BAS

```

DEFDBL A-Z: CLS
OPEN "c:\temp:maj96kc.txt" FOR OUTPUT AS #1
OPTION BASE 0: DIM g(10)
g(0) = .1353352832366127#: g(1) = .2706705664732254#
g(2) = .2706705664732254#: g(3) = .1804470443154836#
g(4) = 9.022352215774179D-02: g(5) = 3.608940886309672D-02
g(6) = 1.202980295436557D-02: g(7) = 3.437086558390164D-03
g(8) = 8.592716395975409D-04: g(9) = 1.90949253243898D-04
FOR i = 0 TO 9: FOR j = 0 TO 9: FOR l = 0 TO 9: FOR m = 0 TO 9:
FOR n = 0 TO 9: FOR o = 0 TO 9: FOR p = 0 TO 9
IF i + j + l + m + n + o + p = 39 THEN
gsum = gsum + g(i) * g(j) * g(l) * g(m) * g(n) * g(o) * g(p)
END IF
NEXT p, o, n, m, l, j, i
tthou = 100000: D1 = CDBL(1 / tthou): D2 = CDBL(2 / tthou)
PRINT #1, " i j l m n o p k f(k) k.f(k)"
PRINT #1, " -----"
FOR i = 0 TO 9: FOR j = 0 TO 9: FOR l = 0 TO 9: FOR m = 0 TO 9:
FOR n = 0 TO 9: FOR o = 0 TO 9: FOR p = 0 TO 9
K = .25
IF i + j + l + m + n + o + p <> 39 THEN GOTO getout1
longloop:
k0 = CDBL(K): k1 = CDBL(K + K * D1): k2 = CDBL(K + K * D2)
K = k0: GOSUB LK0FUNC: L0 = F ' L0 = L(k)
K = k1: GOSUB LK0FUNC: L1 = F ' L1 = L(k+Dk)
K = k2: GOSUB LK0FUNC: L2 = F ' L2 = L(k+2Dk)
K = k0
DLDK = CDBL((L1 - L0) / (K * D1))
D2LDK2 = CDBL((L2 - L1) / (K * D1) - DLDK) / (K * D1)
KONE = CDBL(K - DLDK / D2LDK2)
IF ABS(KONE - K) < 1E-08 THEN GOTO EXACT0 ELSE K = KONE: GOTO longloop
EXACT0:
LOCATE 10

```

```

PRINT USING " ## ## ## ## ## ## ## ## ##.##### "; i; j; l; m; n; o; p; K
PRINT #1, USING " ## ## ## ## ## ## ## ## ##.##### "; i; j; l; m; n; o; p; K
freqk = g(i) * g(j) * g(l) * g(m) * g(n) * g(o) * g(p) / gsum
PRINT #1, USING "##### "; freqk; K * freqk
totk = totk + freqk
ksum = ksum + K * freqk
k2sum = k2sum + K * K * freqk
getout1:
NEXT p, o, n, m, l, j, i
PRINT #1, " -----"
PRINT #1, " ";
PRINT #1, USING "##### "; totk; ksum
kvar = k2sum - ksum * ksum: kse = SQR(kvar)
PRINT #1, USING "k = #.####"; ksum
PRINT #1, USING "se(k) = #.####"; kse
CLOSE #1
END
LK0FUNC:
F = 3 * LOG(K) + 3 * LOG(1 - K) + 3 * LOG(2 - K)
F = F + LOG((3 - 3 * K + K ^ 2) / 3)
F = F + LOG(1 / 2 + (((1 - K) / 2) * ((2 - K) / 2) ^ i))
F = F + LOG(1 / 2 + (((1 - K) / 2) * ((2 - K) / 2) ^ j))
F = F + LOG(1 / 2 + (((1 - K) / 2) * ((2 - K) / 2) ^ l))
F = F + LOG(1 / 2 + (((1 - K) / 2) * ((2 - K) / 2) ^ m))
F = F + LOG(1 / 2 + (((1 - K) / 2) * ((2 - K) / 2) ^ n))
F = F + LOG(1 / 2 + (((1 - K) / 2) * ((2 - K) / 2) ^ o))
F = F + LOG(1 / 2 + (((1 - K) / 2) * ((2 - K) / 2) ^ p))
RETURN

```

MAJ21_2K.BAS

```

DEFDBL A-Z: CLS
OPEN "c:\temp\maj96_2k.txt" FOR OUTPUT AS #1
OPTION BASE 0: DIM g(9)
g(0) = .1353352832366127#: g(1) = .2706705664732254#
g(2) = .2706705664732254#: g(3) = .1804470443154836#
g(4) = 9.022352215774179D-02: g(5) = 3.608940886309672D-02
g(6) = 1.202980295436557D-02: g(7) = 3.437086558390164D-03
g(8) = 8.592716395975409D-04: g(9) = 1.90949253243898D-04
FOR i = 0 TO 9: FOR j = 0 TO 9: FOR l = 0 TO 9: FOR m = 0 TO 9:
FOR n = 0 TO 9: FOR o = 0 TO 9: FOR p = 0 TO 9
IF i + j + l + m + n + o + p = 39 THEN
gsum = gsum + g(i) * g(j) * g(l) * g(m) * g(n) * g(o) * g(p)
END IF
NEXT p, o, n, m, l, j, i
tthou = 100000: D1 = CDBL(1 / tthou): D2 = CDBL(2 / tthou)
PRINT #1, " i j l m n o p k f(k) k.f(k)"
PRINT #1, " -----"
FOR i = 0 TO 9: FOR j = 0 TO 9: FOR l = 0 TO 9: FOR m = 0 TO 9:
FOR n = 0 TO 9: FOR o = 0 TO 9: FOR p = 0 TO 9
K = .25
IF i + j + l + m + n + o + p <> 39 THEN GOTO getout1
longloop:
k0 = CDBL(K): K1 = CDBL(K + K * D1): k2 = CDBL(K + K * D2)
K = k0: GOSUB LK0FUNC: L0 = F ' L0 = L(k)
K = K1: GOSUB LK0FUNC: L1 = F ' L1 = L(k+Dk)
K = k2: GOSUB LK0FUNC: L2 = F ' L2 = L(k+2Dk)
K = k0
DLDK = CDBL((L1 - L0) / (K * D1))
D2LDK2 = CDBL((L2 - L1) / (K * D1) - DLDK) / (K * D1)
KONE = CDBL(K - DLDK / D2LDK2)
IF ABS(KONE - K) < 1E-08 THEN GOTO EXACT0 ELSE K = KONE: GOTO longloop
EXACT0:

```

```

LOCATE 10
PRINT USING " ## ## ## ## ## ## ## ## ##.##### "; i; j; l; m; n; o; p; K
PRINT #1, USING " ## ## ## ## ## ## ## ## ##.##### "; i; j; l; m; n; o; p; K;
freqk = g(i) * g(j) * g(l) * g(m) * g(n) * g(o) * g(p) / gsum
PRINT #1, USING "##### "; freqk; K * freqk
totk = totk + freqk
ksum = ksum + K * freqk
k2sum = k2sum + K * K * freqk
getout1:
NEXT p, o, n, m, l, j, i
PRINT #1, " -----"
PRINT #1, " ";
PRINT #1, USING "##### "; totk; ksum
kvar = k2sum - ksum * ksum: kse = SQR(kvar)
PRINT #1, USING "k = #####"; ksum
PRINT #1, USING "se(k) = #####"; kse
CLOSE #1
END
LK0FUNC:
F = 3 * LOG(K) + 3 * LOG(1) + 3 * LOG(2 - K)
F = F + LOG((3 - 3 * K + K ^ 2) / 3)
F = F + LOG(1 / 2 + ((1 / 2) * ((2 - K) / 2) ^ i))
F = F + LOG(1 / 2 + ((1 / 2) * ((2 - K) / 2) ^ j))
F = F + LOG(1 / 2 + ((1 / 2) * ((2 - K) / 2) ^ l))
F = F + LOG(1 / 2 + ((1 / 2) * ((2 - K) / 2) ^ m))
F = F + LOG(1 / 2 + ((1 / 2) * ((2 - K) / 2) ^ n))
F = F + LOG(1 / 2 + ((1 / 2) * ((2 - K) / 2) ^ o))
F = F + LOG(1 / 2 + ((1 / 2) * ((2 - K) / 2) ^ p))
RETURN

```


X) ANEXO II

LISTAGEM DO PROGRAMA PENCALC V. 1.0, DESENVOLVIDO EM LINGUAGEM MICROSOFT® VISUAL BASIC® 6.0

MÓDULO PUBLIC.BAS

```
Option Explicit
Public resp As String
Public i, j, l, m, n, t, ctr, cont, gem, tela, x, jl, marc, KeyAscii, pen, hetnpen,
norsf, n2g, n2gi(1 To 15), n3g, n3gi(1 To 10), n3gij(1 To 10, 1 To 10), n4g, n4gi(1 To
10), n4gij(1 To 10, 1 To 10), n4gijl(1 To 10, 1 To 10, 1 To 10), n5g, n5gi(1 To 10),
n5gij(1 To 10, 1 To 10), n5gijl(1 To 10, 1 To 10, 1 To 10), n5gijlm(1 To 10, 1 To 10,
1 To 10, 1 To 10), flag(1 To 2), nac, slpen(1 To 2), slhet(1 To 2), slnsf(1 To 2),
s2pen(1 To 2), s2het(1 To 2), s2nsf(1 To 2), sl2g(1 To 2), sl3g(1 To 2), sl4g(1 To 2),
sl5g(1 To 2), s22g(1 To 2), s23g(1 To 2), s24g(1 To 2), s25g(1 To 2), sl2gi(1 To 2, 1
To 15), s22gi(1 To 2, 1 To 15), sl3gi(1 To 2, 1 To 10), s23gi(1 To 2, 1 To 10),
sl3gij(1 To 2, 1 To 10, 1 To 10), s23gij(1 To 2, 1 To 10, 1 To 10), sl4gi(1 To 2, 1 To
10), s24gi(1 To 2, 1 To 10), sl4gij(1 To 2, 1 To 10, 1 To 10), s24gij(1 To 2, 1 To 10,
1 To 10), sl4gijl(1 To 2, 1 To 10, 1 To 10, 1 To 10), s24gijl(1 To 2, 1 To 10, 1 To
10, 1 To 10), sl5gi(1 To 2, 1 To 10), s25gi(1 To 2, 1 To 10), sl5gij(1 To 2, 1 To 10,
1 To 10), s25gij(1 To 2, 1 To 10, 1 To 10), sl5gijl(1 To 2, 1 To 10, 1 To 10, 1 To
10), s25gijl(1 To 2, 1 To 10, 1 To 10, 1 To 10), sl5gijlm(1 To 2, 1 To 10, 1 To 10, 1
To 10, 1 To 10), s25gijlm(1 To 2, 1 To 10, 1 To 10, 1 To 10, 1 To 10, 1 To 10), slgem1(1 To 2),
slgem2(1 To 2), slgem3(1 To 2), s2gem1(1 To 2), s2gem2(1 To 2), s2gem3(1 To 2), ngem1,
ngem2, ngem3 As Integer
Public D1, D2 As Single
Public F1, F2, F3, F4, F5, F6, varkc1, varkc2, varkc3, varkc4, kfinal, k, k0, k01,
k02, kn, kn1, kn2, kc1, kc2, kc3, kc4, k1, k2, F, L0, L1, L2, dLdk, D2LDk2, kone,
totalarea, area, pk, pk0, pk1, h, s, w, niter, plk, p2k, p3k, p4k, p5k, p6k, dif0,
dif1, dif2, lik1, lik2, lik3, lik4, lsk1, lsk2, lsk3, lsk4, lifinal, lsfinal As Double
```

Public Sub ReadFile()

```
Input #1, pen: Input #1, hetnpen: Input #1, norsf
Input #1, n2g
For i = 1 To n2g: Input #1, n2gi(i): Next i
Input #1, n3g
For i = 1 To n3g
    Input #1, n3gi(i)
    For j = 1 To n3gi(i): Input #1, n3gij(i, j): Next j
Next i
Input #1, n4g
For i = 1 To n4g
    Input #1, n4gi(i)
    For j = 1 To n4gi(i)
        Input #1, n4gij(i, j)
        For l = 1 To n4gij(i, j): Input #1, n4gijl(i, j, l): Next l
    Next j
Next i
Input #1, n5g
For i = 1 To n5g
    Input #1, n5gi(i)
    For j = 1 To n5gi(i)
        Input #1, n5gij(i, j)
        For l = 1 To n5gij(i, j)
            Input #1, n5gijl(i, j, l)
            For m = 1 To n5gijl(i, j, l): Input #1, n5gijlm(i, j, l, m): Next m
        Next l
    Next j
Next i
Input #1, ngem1: Input #1, ngem2: Input #1, ngem3
Input #1, nac
For x = 1 To nac
    Input #1, slpen(x): Input #1, slhet(x): Input #1, slnsf(x)
    Input #1, sl2g(x)
    For i = 1 To sl2g(x): Input #1, sl2gi(x, i): Next i
    Input #1, sl3g(x)
    For i = 1 To sl3g(x)
        Input #1, sl3gi(x, i)
        For j = 1 To sl3gi(x, i): Input #1, sl3gij(x, i, j): Next j
    Next i
    Input #1, sl4g(x)
```

```

For i = 1 To s14g(x)
    Input #1, s14gi(x, i)
    For j = 1 To s14gi(x, i)
        Input #1, s14gij(x, i, j)
        For l = 1 To s14gij(x, i, j): Input #1, s14gijl(x, i, j, l): Next l
    Next j
Next i
Input #1, s15g(x)
For i = 1 To s15g(x)
    Input #1, s15gi(x, i)
    For j = 1 To s15gi(x, i)
        Input #1, s15gij(x, i, j)
        For l = 1 To s15gij(x, i, j)
            Input #1, s15gijl(x, i, j, l)
            For m = 1 To s15gijl(x, i, j, l): Input #1, s15gijlm(x, i, j, l, m): Next m
        Next l
    Next j
Next i
Input #1, slgem1(x): Input #1, slgem2(x): Input #1, slgem3(x)
Input #1, s2pen(x):
If s2pen(x) = 99 Then
    flag(x) = 0
    s2pen(x) = s1pen(x): s2het(x) = slhet(x): s2nsf(x) = slnsf(x)
    s22g(x) = s12g(x)
    For i = 1 To s12g(x): s22gi(x, i) = s12gi(x, i): Next i
    s23g(x) = s13g(x)
    For i = 1 To s13g(x)
        s23gi(x, i) = s13gi(x, i)
        For j = 1 To s13gi(x, i): s23gij(x, i, j) = s13gij(x, i, j): Next j
    Next i
    s24g(x) = s14g(x)
    For i = 1 To s14g(x)
        s24gi(x, i) = s14gi(x, i)
        For j = 1 To s14gi(x, i)
            s24gij(x, i, j) = s14gij(x, i, j)
            For l = 1 To s14gij(x, i, j): s24gijl(x, i, j, l) = s14gijl(x, i, j, l):
Next l
        Next j
    Next i
    s25g(x) = s15g(x)
    For i = 1 To s15g(x)
        s25gi(x, i) = s15gi(x, i)
        For j = 1 To s15gi(x, i)
            s25gij(x, i, j) = s15gij(x, i, j)
            For l = 1 To s15gij(x, i, j)
                s25gijl(x, i, j, l) = s15gijl(x, i, j, l)
                For m = 1 To s15gijl(x, i, j, l): s25gijlm(x, i, j, l, m) = s15gijlm(x, i,
j, l, m): Next m
            Next l
        Next j
    Next i
    s2gem1(x) = slgem1(x): s2gem2(x) = slgem2(x): s2gem3(x) = slgem3(x)
Else
    flag(x) = 1
    Input #1, s2het(x): Input #1, s2nsf(x)
    Input #1, s22g(x)
    For i = 1 To s22g(x): Input #1, s22gi(x, i): Next i
    Input #1, s23g(x)
    For i = 1 To s23g(x)
        Input #1, s23gi(x, i)
        For j = 1 To s23gi(x, i): Input #1, s23gij(x, i, j): Next j
    Next i
    Input #1, s24g(x)
    For i = 1 To s24g(x)
        Input #1, s24gi(x, i)
        For j = 1 To s24gi(x, i)
            Input #1, s24gij(x, i, j)
            For l = 1 To s24gij(x, i, j): Input #1, s24gijl(x, i, j, l): Next l
        Next j
    Next i
    Input #1, s25g(x)
    For i = 1 To s25g(x)
        Input #1, s25gi(x, i)
        For j = 1 To s25gi(x, i)
            Input #1, s25gij(x, i, j)
            For l = 1 To s25gij(x, i, j)
                Input #1, s25gijl(x, i, j, l)

```

```

        For m = 1 To s25gijl(x, i, j, 1): Input #1, s25gijlm(1, i, j, 1, m): Next
m
        Next l
        Next j
        Next i
        Input #1, s2gem1(x): Input #1, s2gem2(x): Input #1, s2gem3(x)
        End If
        Next x
        Close #1
End Sub

```

```

Public Sub Run()
    frm_inicial.Label10.Visible = True: frm_inicial.txt_ident.Visible = True
    frm_inicial.mnuFilePrint.Enabled = True
    frm_inicial.mnuFileSaveas.Enabled = True
    marc = 0
    10 k = 0.5
    LOOP1:
        If k <= 0 Then k = 0.05
        If k >= 1 Then k = 0.95
        D1 = 0.00001: D2 = 0.00002
        k0 = k: k1 = k + k * D1: k2 = k + k * D2
        k = k0: GoSub Func: L0 = F ' L0 = L(k)
        k = k1: GoSub Func: L1 = F ' L1 = L(k+Dk)
        k = k2: GoSub Func: L2 = F ' L2 = L(k+2Dk)
        k = k0
        dLdk = (L1 - L0) / (k * D1)
        D2LDk2 = ((L2 - L1) / (k * D1) - dLdk) / (k * D1)
        kone = k - dLdk / D2LDk2
        If Abs(kone - k) < 0.0000001 Then
            frm_inicial.txt_ident.SetFocus
            frm_inicial.Label6.Visible = True: frm_inicial.txt_kvalue.Visible = True
            If nac <> 0 Then
                If flag(1) = 0 Or flag(2) = 0 Or nac = 1 Then
                    frm_inicial.txt_k1.Visible = True: frm_inicial.txt_k2.Visible = True
                    frm_inicial.Label12(0).Visible = True
                    frm_inicial.Label12(1).Visible = True
                    frm_inicial.Refresh
                    If marc = 0 Then 'S1
                        kc1 = kone: varkc1 = -1 / D2LDk2
                        frm_inicial.txt_k1.Text = Format$(kc1, "0.000"): marc = 1: GoTo 10
                    ElseIf marc = 1 Then 'S2
                        kc2 = kone: varkc2 = -1 / D2LDk2
                        frm_inicial.txt_k2.Text = Format$(kc2, "0.000")
                        kfinal = (kc1 / varkc1 + kc2 / varkc2) / (1 / varkc1 + 1 / varkc2)
                        frm_inicial.txt_kvalue.Text = Format$(kfinal, "0.000")
                    End If
                End If
                If flag(1) = 1 And flag(2) = 1 And nac = 2 Then
                    frm_inicial.txt_k1.Visible = True: frm_inicial.txt_k2.Visible = True
                    frm_inicial.txt_kc2.Visible = True: frm_inicial.txt_kc4.Visible = True
                    For i = 1 To 4: frm_inicial.Label9(i - 1).Visible = True: Next i
                    frm_inicial.Refresh
                    If marc = 0 Then 'duas S1
                        kc1 = kone: varkc1 = -1 / D2LDk2
                        frm_inicial.txt_k1.Text = Format$(kc1, "0.000"): marc = 1: GoTo 10
                    ElseIf marc = 1 Then 'duas S2
                        kc2 = kone: varkc2 = -1 / D2LDk2
                        frm_inicial.txt_kc2.Text = Format$(kc2, "0.000"): marc = 2: GoTo 10
                    ElseIf marc = 2 Then 'S1 com S2
                        kc3 = kone: varkc3 = -1 / D2LDk2
                        frm_inicial.txt_k2.Text = Format$(kc3, "0.000"): marc = 3: GoTo 10
                    ElseIf marc = 3 Then 'S2 com S1
                        kc4 = kone: varkc4 = -1 / D2LDk2
                        frm_inicial.txt_kc4.Text = Format$(kc4, "0.000")
                        kfinal = kc1 / varkc1 + kc2 / varkc2 + kc3 / varkc3 + kc4 / varkc4
                        kfinal = kfinal / (1 / varkc1 + 1 / varkc2 + 1 / varkc3 + 1 / varkc4)
                        frm_inicial.txt_kvalue.Text = Format$(kfinal, "0.000")
                    End If
                End If
            Else: frm_inicial.txt_kvalue.Text = Format$(kone, "0.000")
            End If
            frm_inicial.Refresh
            Call WriteFunction
            Call Confint
            frm_inicial.txt_ident.SetFocus

```

```

        GoTo endprogram
    Else: k = kone: GoTo LOOP1
End If
Func:
F = pen * Log(k) + hetnpen * Log(1 - k) + norsf * Log(2 - k)
For i = 1 To n2g
    F = F + Log(1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ n2gi(i)):
Next i
For i = 1 To n3g
    F1 = 1
    For j = 1 To n3gi(i)
        F1 = F1 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ n3gij(i, j))
    Next j
    F1 = F1 * (1 - k) / 2 + 1 / 2: F = F + Log(F1)
Next i
For i = 1 To n4g
    F2 = 1: F3 = 1
    For j = 1 To n4gi(i)
        For l = 1 To n4gij(i, j)
            F2 = F2 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ n4gijl(i, j, l))
        Next l
        F3 = F3 * (F2 * (1 - k) / 2 + 1 / 2): F2 = 1
    Next j
    F3 = F3 * (1 - k) / 2 + 1 / 2: F = F + Log(F3)
Next i
For i = 1 To n5g
    F4 = 1: F5 = 1: F6 = 1
    For j = 1 To n5gi(i)
        For l = 1 To n5gij(i, j)
            For m = 1 To n5gijl(i, j, l)
                F5 = F5 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ n5gijlm(i, j, l, m))
            Next m
            F6 = F6 * (F5 * (1 - k) / 2 + 1 / 2): F5 = 1
        Next l
        F4 = F4 * (F6 * (1 - k) / 2 + 1 / 2): F6 = 1
    Next j
    F4 = F4 * (1 - k) / 2 + 1 / 2: F = F + Log(F4)
Next i
For x = 1 To nac
    If marc = 0 Then
        F = F + slpen(x) * Log(k) + slhet(x) * Log(1 - k) + slnsf(x) * Log(2 - k)
        For i = 1 To sl2g(x)
            F = F + Log(1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl2gi(x, i))
        Next i
        For i = 1 To sl3g(x)
            F1 = 1
            For j = 1 To sl3gi(x, i)
                F1 = F1 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl3gij(x, i, j))
            Next j
            F1 = F1 * (1 - k) / 2 + 1 / 2: F = F + Log(F1)
        Next i
        For i = 1 To sl4g(x)
            F2 = 1: F3 = 1
            For j = 1 To sl4gi(x, i)
                For l = 1 To sl4gij(x, i, j)
                    F2 = F2 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl4gijl(x, i, j, l))
                Next l
                F3 = F3 * (F2 * (1 - k) / 2 + 1 / 2): F2 = 1
            Next j
            F3 = F3 * (1 - k) / 2 + 1 / 2: F = F + Log(F3)
        Next i
        For i = 1 To sl5g(x)
            F4 = 1: F5 = 1: F6 = 1
            For j = 1 To sl5gi(x, i)
                For l = 1 To sl5gij(x, i, j)
                    For m = 1 To sl5gijl(x, i, j, l)
                        F5 = F5 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl5gijlm(x, i, j, l,
m))
                    Next m
                    F6 = F6 * (F5 * (1 - k) / 2 + 1 / 2): F5 = 1
                Next l
                F4 = F4 * (F6 * (1 - k) / 2 + 1 / 2): F6 = 1
            Next j
            F4 = F4 * (1 - k) / 2 + 1 / 2: F = F + Log(F4)
        Next i
        For i = 1 To slgem1(x): F = F + Log((3 - 3 * k + k ^ 2) / 3): Next i
        For i = 1 To slgem2(x): F = F + Log(k * (3 - 2 * k) / 6): Next i
    End If
Next x

```

```

For i = 1 To slgem3(x): F = F + Log(k ^ 2 / 3): Next i
ElseIf marc = 1 Then
F = F + s2pen(x) * Log(k) + s2het(x) * Log(1 - k) + s2nsf(x) * Log(2 - k)
For i = 1 To s22g(x)
F = F + Log(1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s22gi(x, i))
Next i
For i = 1 To s23g(x)
F1 = 1
For j = 1 To s23gi(x, i)
F1 = F1 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s23gij(x, i, j))
Next j
F1 = F1 * (1 - k) / 2 + 1 / 2: F = F + Log(F1)
Next i
For i = 1 To s24g(x)
F2 = 1: F3 = 1
For j = 1 To s24gi(x, i)
For l = 1 To s24gij(x, i, j)
F2 = F2 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s24gijl(x, i, j, l))
Next l
F3 = F3 * (F2 * (1 - k) / 2 + 1 / 2): F2 = 1
Next j
F3 = F3 * (1 - k) / 2 + 1 / 2: F = F + Log(F3)
Next i
For i = 1 To s25g(x)
F4 = 1: F5 = 1: F6 = 1
For j = 1 To s25gi(x, i)
For l = 1 To s25gij(x, i, j)
For m = 1 To s25gijl(x, i, j, l)
F5 = F5 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s25gijlm(x, i, j, l,
m))
Next m
F6 = F6 * (F5 * (1 - k) / 2 + 1 / 2): F5 = 1
Next l
F4 = F4 * (F6 * (1 - k) / 2 + 1 / 2): F6 = 1
Next j
F4 = F4 * (1 - k) / 2 + 1 / 2: F = F + Log(F4)
Next i
For i = 1 To s2gem1(x): F = F + Log((3 - 3 * k + k ^ 2) / 3): Next i
For i = 1 To s2gem2(x): F = F + Log(k * (3 - 2 * k) / 6): Next i
For i = 1 To s2gem3(x): F = F + Log(k ^ 2 / 3): Next i
ElseIf marc = 2 Then
If x = 1 Then
F = F + slpen(x) * Log(k) + slhet(x) * Log(1 - k) + slnsf(x) * Log(2 - k)
For i = 1 To sl2g(x)
F = F + Log(1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl2gi(x, i))
Next i
For i = 1 To sl3g(x)
F1 = 1
For j = 1 To sl3gi(x, i)
F1 = F1 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl3gij(x, i, j))
Next j
F1 = F1 * (1 - k) / 2 + 1 / 2: F = F + Log(F1)
Next i
For i = 1 To sl4g(x)
F2 = 1: F3 = 1
For j = 1 To sl4gi(x, i)
For l = 1 To sl4gij(x, i, j)
F2 = F2 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl4gijl(x, i, j, l))
Next l
F3 = F3 * (F2 * (1 - k) / 2 + 1 / 2): F2 = 1
Next j
F3 = F3 * (1 - k) / 2 + 1 / 2: F = F + Log(F3)
Next i
For i = 1 To sl5g(x)
F4 = 1: F5 = 1: F6 = 1
For j = 1 To sl5gi(x, i)
For l = 1 To sl5gij(x, i, j)
For m = 1 To sl5gijl(x, i, j, l)
F5 = F5 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl5gijlm(x, i, j, l,
m))
Next m
F6 = F6 * (F5 * (1 - k) / 2 + 1 / 2): F5 = 1
Next l
F4 = F4 * (F6 * (1 - k) / 2 + 1 / 2): F6 = 1
Next j
F4 = F4 * (1 - k) / 2 + 1 / 2: F = F + Log(F4)
Next i

```

```

For i = 1 To slgem1(x): F = F + Log((3 - 3 * k + k ^ 2) / 3): Next i
For i = 1 To slgem2(x): F = F + Log(k * (3 - 2 * k) / 6): Next i
For i = 1 To slgem3(x): F = F + Log(k ^ 2 / 3): Next i
ElseIf x = 2 Then
F = F + s2pen(x) * Log(k) + s2het(x) * Log(1 - k) + s2nsf(x) * Log(2 - k)
For i = 1 To s22g(x)
F = F + Log(1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s22gi(x, i))
Next i
For i = 1 To s23g(x)
F1 = 1
For j = 1 To s23gi(x, i)
F1 = F1 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s23gij(x, i, j))
Next j
F1 = F1 * (1 - k) / 2 + 1 / 2: F = F + Log(F1)
Next i
For i = 1 To s24g(x)
F2 = 1: F3 = 1
For j = 1 To s24gi(x, i)
For l = 1 To s24gij(x, i, j)
F2 = F2 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s24gijl(x, i, j, l))
Next l
F3 = F3 * (F2 * (1 - k) / 2 + 1 / 2): F2 = 1
Next j
F3 = F3 * (1 - k) / 2 + 1 / 2: F = F + Log(F3)
Next i
For i = 1 To s25g(x)
F4 = 1: F5 = 1: F6 = 1
For j = 1 To s25gi(x, i)
For l = 1 To s25gij(x, i, j)
For m = 1 To s25gijl(x, i, j, l)
F5 = F5 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s25gijlm(x, i, j, l,
m))
Next m
F6 = F6 * (F5 * (1 - k) / 2 + 1 / 2): F5 = 1
Next l
F4 = F4 * (F6 * (1 - k) / 2 + 1 / 2): F6 = 1
Next j
F4 = F4 * (1 - k) / 2 + 1 / 2: F = F + Log(F4)
Next i
For i = 1 To s2gem1(x): F = F + Log((3 - 3 * k + k ^ 2) / 3): Next i
For i = 1 To s2gem2(x): F = F + Log(k * (3 - 2 * k) / 6): Next i
For i = 1 To s2gem3(x): F = F + Log(k ^ 2 / 3): Next i
End If
ElseIf marc = 3 Then
If x = 1 Then
F = F + s2pen(x) * Log(k) + s2het(x) * Log(1 - k) + s2nsf(x) * Log(2 - k)
For i = 1 To s22g(x)
F = F + Log(1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s22gi(x, i))
Next i
For i = 1 To s23g(x)
F1 = 1
For j = 1 To s23gi(x, i)
F1 = F1 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s23gij(x, i, j))
Next j
F1 = F1 * (1 - k) / 2 + 1 / 2: F = F + Log(F1)
Next i
For i = 1 To s24g(x)
F2 = 1: F3 = 1
For j = 1 To s24gi(x, i)
For l = 1 To s24gij(x, i, j)
F2 = F2 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s24gijl(x, i, j, l))
Next l
F3 = F3 * (F2 * (1 - k) / 2 + 1 / 2): F2 = 1
Next j
F3 = F3 * (1 - k) / 2 + 1 / 2: F = F + Log(F3)
Next i
For i = 1 To s25g(x)
F4 = 1: F5 = 1: F6 = 1
For j = 1 To s25gi(x, i)
For l = 1 To s25gij(x, i, j)
For m = 1 To s25gijl(x, i, j, l)
F5 = F5 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s25gijlm(x, i, j, l,
m))
Next m
F6 = F6 * (F5 * (1 - k) / 2 + 1 / 2): F5 = 1
Next l
F4 = F4 * (F6 * (1 - k) / 2 + 1 / 2): F6 = 1

```

```

        Next j
        F4 = F4 * (1 - k) / 2 + 1 / 2: F = F + Log(F4)
    Next i
    For i = 1 To s2gem1(x): F = F + Log((3 - 3 * k + k ^ 2) / 3): Next i
    For i = 1 To s2gem2(x): F = F + Log(k * (3 - 2 * k) / 6): Next i
    For i = 1 To s2gem3(x): F = F + Log(k ^ 2 / 3): Next i
ElseIf x = 2 Then
    F = F + slpen(x) * Log(k) + slhet(x) * Log(1 - k) + slnsf(x) * Log(2 - k)
    For i = 1 To sl2g(x)
        F = F + Log(1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl2gi(x, i))
    Next i
    For i = 1 To sl3g(x)
        F1 = 1
        For j = 1 To sl3gi(x, i)
            F1 = F1 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl3gij(x, i, j))
        Next j
        F1 = F1 * (1 - k) / 2 + 1 / 2: F = F + Log(F1)
    Next i
    For i = 1 To sl4g(x)
        F2 = 1: F3 = 1
        For j = 1 To sl4gi(x, i)
            For l = 1 To sl4gij(x, i, j)
                F2 = F2 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl4gijl(x, i, j, l))
            Next l
            F3 = F3 * (F2 * (1 - k) / 2 + 1 / 2): F2 = 1
        Next j
        F3 = F3 * (1 - k) / 2 + 1 / 2: F = F + Log(F3)
    Next i
    For i = 1 To sl5g(x)
        F4 = 1: F5 = 1: F6 = 1
        For j = 1 To sl5gi(x, i)
            For l = 1 To sl5gij(x, i, j)
                For m = 1 To sl5gijl(x, i, j, l)
                    F5 = F5 * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl5gijlm(x, i, j, l,
m))
                Next m
                F6 = F6 * (F5 * (1 - k) / 2 + 1 / 2): F5 = 1
            Next l
            F4 = F4 * (F6 * (1 - k) / 2 + 1 / 2): F6 = 1
        Next j
        F4 = F4 * (1 - k) / 2 + 1 / 2: F = F + Log(F4)
    Next i
    For i = 1 To slgem1(x): F = F + Log((3 - 3 * k + k ^ 2) / 3): Next i
    For i = 1 To slgem2(x): F = F + Log(k * (3 - 2 * k) / 6): Next i
    For i = 1 To slgem3(x): F = F + Log(k ^ 2 / 3): Next i
End If
End If
Next x
For i = 1 To ngem1: F = F + Log((3 - 3 * k + k ^ 2) / 3): Next i
For i = 1 To ngem2: F = F + Log(k * (3 - 2 * k) / 6): Next i
For i = 1 To ngem3: F = F + Log(k ^ 2 / 3): Next i
Return
endprogram:
frm_inicial.Visible = True
End Sub

```

```

Public Sub WriteFunction()
    Open App.Path & "\function.txt" For Output As #2
    If n2g <> 0 Or n3g <> 0 Or n4g <> 0 Or n5g <> 0 Or nac <> 0 Then
        Print #2, "F = K^"; pen & " . (1-K)^"; hetnpen & " . (2-K)^"; norsf & " . "
        If ngem1 <> 0 Then
            If ngem2 <> 0 And ngem3 <> 0 Then Print #2, "      (3-3K+K^2)^"; ngem1 & " . "
[K*(3-2K)]^"; ngem2 & " . (K^2)^"; ngem3 & " . "
            If ngem2 = 0 And ngem3 = 0 Then Print #2, "      (3-3K+K^2)^"; ngem1 & " . "
            If ngem2 <> 0 And ngem3 = 0 Then Print #2, "      (3-3K+K^2)^"; ngem1 & " . [K*(3-
2K)]^"; ngem2 & " . "
            If ngem2 = 0 And ngem3 <> 0 Then Print #2, "      (3-3K+K^2)^"; ngem1 & " . "
(K^2)^"; ngem3 & " . "
            ElseIf ngem1 = 0 And ngem2 <> 0 Then
                If ngem3 = 0 Then Print #2, "      [K*(3-2K)]^"; ngem2 & " . "
                If ngem3 <> 0 Then Print #2, "      [K*(3-2K)]^"; ngem2 & " . (K^2)^"; ngem3 & " . "
            "
            ElseIf ngem1 = 0 And ngem2 = 0 And ngem3 <> 0 Then Print #2, "      (K^2)^"; ngem3 &
" . "
            End If
            ElseIf ngem1 <> 0 Or ngem2 <> 0 Or ngem3 <> 0 Then

```

```

Print #2, "F = K^"; pen & " . (1-K)^"; hetnpen & " . (2-K)^"; norsf & " . "
If ngem1 <> 0 Then
  If ngem2 <> 0 And ngem3 <> 0 Then Print #2, "      (3-3K+K^2)^"; ngem1 & " .
[K*(3-2K)]^"; ngem2 & " . (K^2)^"; ngem3 & " "
  If ngem2 = 0 And ngem3 = 0 Then Print #2, "      (3-3K+K^2)^"; ngem1 & " "
  If ngem2 <> 0 And ngem3 = 0 Then Print #2, "      (3-3K+K^2)^"; ngem1 & " . [K*(3-
2K)]^"; ngem2 & " "
  If ngem2 = 0 And ngem3 <> 0 Then Print #2, "      (3-3K+K^2)^"; ngem1 & " .
(K^2)^"; ngem3 & " "
  ElseIf ngem1 = 0 And ngem2 <> 0 Then
    If ngem3 = 0 Then Print #2, "      [K*(3-2K)]^"; ngem2 & " "
    If ngem3 <> 0 Then Print #2, "      [K*(3-2K)]^"; ngem2 & " . (K^2)^"; ngem3 & " "
  ElseIf ngem1 = 0 And ngem2 = 0 And ngem3 <> 0 Then Print #2, "      (K^2)^"; ngem3 &
" "
  End If
Else: Print #2, "F = K^"; pen & " . (1-K)^"; hetnpen & " . (2-K)^"; norsf & " "
End If
For i = 1 To n2g
  If n3g <> 0 Or n4g <> 0 Or n5g <> 0 Or nac <> 0 Then
    Print #2, "      {1/2 + (1-K)/2.[(2-K)/2]^"; n2gi(i) & " } ."
  Else:
    If i = n2g Then Print #2, "      {1/2 + (1-K)/2.[(2-K)/2]^"; n2gi(i) & " }": GoTo
10
    Print #2, "      {1/2 + (1-K)/2.[(2-K)/2]^"; n2gi(i) & " } ."
  End If
Next i
10 For i = 1 To n3g
  Print #2, "      {1/2 + (1-K)/2.";
  For j = 1 To n3gi(i)
    If n4g <> 0 Or n5g <> 0 Or nac <> 0 Or i < n3g Then
      If n3gi(i) = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n3gij(i, j) & " }"
      .": GoTo 19
      If j = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n3gij(i, j) & " } ."
      If j > 1 And j < n3gi(i) Then Print #2, "      {1/2 + (1-
K)/2.[(2-K)/2]^"; n3gij(i, j) & " } ."
      If j = n3gi(i) Then Print #2, "      {1/2 + (1-K)/2.[(2-
K)/2]^"; n3gij(i, j) & " }"
    Else
      If n3gi(i) = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n3gij(i, j) &
"}": GoTo 19
      If j = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n3gij(i, j) & " } ."
      If j > 1 And j < n3gi(i) Then Print #2, "      {1/2 + (1-
K)/2.[(2-K)/2]^"; n3gij(i, j) & " } ."
      If j = n3gi(i) Then Print #2, "      {1/2 + (1-K)/2.[(2-
K)/2]^"; n3gij(i, j) & " }"
    End If
  Next j
19 Next i
20 For i = 1 To n4g
  Print #2, "      {1/2 + (1-K)/2.";
  For j = 1 To n4gi(i)
    If j = 1 And n4gij(i, j) <> 0 Then Print #2, "{1/2 + (1-K)/2.";
    If j = 1 And n4gij(i, j) = 0 Then Print #2, "[ (2-K)/2] ."
    If j > 1 And n4gij(i, j) <> 0 Then Print #2, "      {1/2 + (1-
K)/2.";
    If j > 1 And n4gij(i, j) = 0 Then Print #2, "      [ (2-K)/2] ."
    If j = n4gi(i) And j <> 1 And n4gij(i, j) = 0 Then Print #2, "
[ (2-K)/2]] "
    For l = 1 To n4gij(i, j)
      If n5g <> 0 Or nac <> 0 Or i < n4g Then
        If n4gi(i) = 1 And n4gij(i, j) = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-
K)/2]^"; n4gijl(i, j, l) & " }" . ": GoTo 29
        If l = 1 And l <> n4gij(i, j) Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^";
n4gijl(i, j, l) & " } ."
        If l = 1 And l = n4gij(i, j) And j <> n4gi(i) Then Print #2, "{1/2 + (1-
K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & " } ."
        If l > 1 And l <> n4gij(i, j) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & " } ."
        If l = n4gij(i, j) And l <> 1 And j <> n4gi(i) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & " } ."
        If l = n4gij(i, j) And l = 1 And j = n4gi(i) Then Print #2, "{1/2 + (1-
K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & " }"
        If l = n4gij(i, j) And l <> 1 And j = n4gi(i) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & " }" .": GoTo 29
      Else:
        If n4gi(i) = 1 And n4gij(i, j) = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-
K)/2]^"; n4gijl(i, j, l) & " }" .": GoTo 29

```



```

        If l = 1 And l <> n4gij(i, j) Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^";
n4gijl(i, j, l) & "}" . "
        If l = 1 And l = n4gij(i, j) And j <> n4gi(i) Then Print #2, "{1/2 + (1-
K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & "}" . "
        If l > 1 And l <> n4gij(i, j) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & "}" . "
        If l = n4gij(i, j) And l <> 1 And j <> n4gi(i) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & "}" . "
        If l = n4gij(i, j) And l = 1 And j = n4gi(i) Then Print #2, "{1/2 + (1-
K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & "}" . "
        If l = n4gij(i, j) And l <> 1 And j = n4gi(i) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n4gijl(i, j, l) & "}" . "
    End If
Next l
Next j
29 Next i
30 For i = 1 To n5g
    Print #2, "    {1/2 + (1-K)/2.";
    For j = 1 To n5gi(i)
        If j = 1 And n5gij(i, j) <> 0 Then Print #2, "{1/2 + (1-K)/2.";
        If j = 1 And n5gij(i, j) = 0 Then Print #2, "[ (2-K)/2] . "
        If j > 1 And n5gij(i, j) <> 0 Then Print #2, "    {1/2 + (1-
K)/2.";
        If j > 1 And j <> n5gi(i) And n5gij(i, j) = 0 Then Print #2, "
[ (2-K)/2] . "
        If j = n5gi(i) And j <> 1 And n5gij(i, j) = 0 And nac = 0 Then Print #2, "
[ (2-K)/2] "
        If j = n5gi(i) And j <> 1 And n5gij(i, j) = 0 And nac <> 0 Then Print #2, "
[ (2-K)/2] . "
        For l = 1 To n5gij(i, j)
            If l = 1 And n5gijl(i, j, l) <> 0 Then Print #2, "{1/2 + (1-K)/2.";
            If l = 1 And l <> n5gij(i, j) And n5gijl(i, j, l) = 0 Then Print #2, "[ (2-
K)/2] . "
            If l > 1 And n5gijl(i, j, l) <> 0 Then Print #2, "
{1/2 + (1-K)/2.";
            If l > 1 And l <> n5gij(i, j) And n5gijl(i, j, l) = 0 Then Print #2, "
[ (2-K)/2] . "
            If l > 1 And l = n5gij(i, j) And n5gijl(i, j, l) = 0 And j <> n5gi(i) Then
Print #2, "    [(2-K)/2] . "
            If l > 1 And l = n5gij(i, j) And n5gijl(i, j, l) = 0 And j = n5gi(i) Then
Print #2, "    [(2-K)/2] "
            For m = 1 To n5gijl(i, j, l)
                If i < n5g Or nac <> 0 Then
                    If n5gi(i) = 1 And n5gij(i, j) = 1 And n5gijl(i, j, l) = 1 Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . " : GoTo 39
                    If l = 1 And l <> n5gij(i, j) And m = 1 And m <> n5gijl(i, j, l) Then
Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l = 1 And l <> n5gij(i, j) And m > 1 And m <> n5gijl(i, j, l) Then
Print #2, "    {1/2 + (1-K)/2.[(2-
K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l = 1 And l <> n5gij(i, j) And m = n5gijl(i, j, l) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l > 1 And l <> n5gij(i, j) And m = 1 And m <> n5gijl(i, j, l) Then
Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l > 1 And l <> n5gij(i, j) And m > 1 And m <> n5gijl(i, j, l) Then
Print #2, "    {1/2 + (1-K)/2.[(2-
K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l > 1 And l <> n5gij(i, j) And m = n5gijl(i, j, l) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l = n5gij(i, j) And m = 1 And m <> n5gijl(i, j, l) Then Print #2,
"{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l = n5gij(i, j) And m > 1 And m <> n5gijl(i, j, l) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l = n5gij(i, j) And m = n5gijl(i, j, l) And j <> n5gi(i) Then Print
#2, "    {1/2 + (1-K)/2.[(2-K)/2]^";
n5gijlm(i, j, l, m) & "}" . "
                    If l = n5gij(i, j) And m = n5gijl(i, j, l) And j = n5gi(i) Then Print
#2, "    {1/2 + (1-K)/2.[(2-K)/2]^";
n5gijlm(i, j, l, m) & "}" . "
                Else:
                    If n5gi(i) = 1 And n5gij(i, j) = 1 And n5gijl(i, j, l) = 1 Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . " : GoTo 39
                    If l = 1 And l <> n5gij(i, j) And m = 1 And m <> n5gijl(i, j, l) Then
Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}" . "
                    If l = 1 And l <> n5gij(i, j) And m > 1 And m <> n5gijl(i, j, l) Then
Print #2, "    {1/2 + (1-K)/2.[(2-
K)/2]^"; n5gijlm(i, j, l, m) & "}" . "

```

```

        If l = 1 And l <> n5gij(i, j) And m = n5gijl(i, j, l) Then Print #2, "
        {1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}} . "
        If l > 1 And l <> n5gij(i, j) And m = 1 And m <> n5gijl(i, j, l) Then
Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}} . "
        If l > 1 And l <> n5gij(i, j) And m > 1 And m <> n5gijl(i, j, l) Then
Print #2, "
        {1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}} . "
        If l > 1 And l <> n5gij(i, j) And m = n5gijl(i, j, l) Then Print #2, "
        {1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}} . "
        If l = n5gij(i, j) And m = 1 And m <> n5gijl(i, j, l) Then Print #2,
        "{1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}} . "
        If l = n5gij(i, j) And m > 1 And m <> n5gijl(i, j, l) Then Print #2, "
        {1/2 + (1-K)/2.[(2-K)/2]^"; n5gijlm(i, j, l, m) & "}} . "
        If l = n5gij(i, j) And m = n5gijl(i, j, l) And j <> n5gi(i) Then Print
        {1/2 + (1-K)/2.[(2-K)/2]^";
n5gijlm(i, j, l, m) & "}} . "
        If l = n5gij(i, j) And m = n5gijl(i, j, l) And j = n5gi(i) Then Print
        {1/2 + (1-K)/2.[(2-K)/2]^";
n5gijlm(i, j, l, m) & "}} . "
        End If
    Next m
Next l
Next j
39 Next i
    For x = 1 To nac
    Print #2,
    Print #2, "    CONSANGUINEOUS TREE "; x
    Print #2, "    Configuration 1:"
    If slpen(x) <> 0 Or slhet(x) <> 0 Or slnsf(x) <> 0 Then
        If x = nac And flag(x) = 0 Then
            Print #2, "    K^"; slpen(x) & " . (1-K)^"; slhet(x) & " . (2-K)^";
slnsf(x) & " . "
        Else:
            Print #2, "    K^"; slpen(x) & " . (1-K)^"; slhet(x) & " . (2-K)^";
slnsf(x) & " . "
        End If
    End If
    For i = 1 To sl2g(x)
        If x = nac And flag(x) = 0 And sl3g(x) = 0 And sl4g(x) = 0 And sl5g(x) = 0 Then
            If i < sl2g(x) Then Print #2, "    {1/2 + (1-K)/2.[(2-K)/2]^"; sl2gi(x, i)
& "}} . "
            If i = sl2g(x) Then Print #2, "    {1/2 + (1-K)/2.[(2-K)/2]^"; sl2gi(x, i)
& "}} . "
        Else:
            Print #2, "    {1/2 + (1-K)/2.[(2-K)/2]^"; sl2gi(x, i) & "}} . "
        End If
    Next i
    For i = 1 To sl3g(x)
        Print #2, "    {1/2 + (1-K)/2.[";
        For j = 1 To sl3gi(x, i)
            If x = nac And flag(x) = 0 And sl4g(x) = 0 And sl5g(x) = 0 Then
                If sl3gi(x, i) = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; sl3gij(x, i,
j) & "}}": GoTo 49
                If j = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; sl3gij(x, i, j) & "}}
. "
                If j > 1 And j < sl3gi(x, i) Then Print #2, "    {1/2 +
(1-K)/2.[(2-K)/2]^"; sl3gij(x, i, j) & "}} . "
                If j = sl3gi(x, i) And i < sl3g(x) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; sl3gij(x, i, j) & "}} . "
                If j = sl3gi(x, i) And i = sl3g(x) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; sl3gij(x, i, j) & "}} . "
            Else
                If sl3gi(x, i) = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; sl3gij(x, i,
j) & "}} .": GoTo 49
                If j = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; sl3gij(x, i, j) & "}}
. "
                If j > 1 And j < sl3gi(x, i) Then Print #2, "    {1/2 +
(1-K)/2.[(2-K)/2]^"; sl3gij(x, i, j) & "}} . "
                If j = sl3gi(x, i) Then Print #2, "    {1/2 + (1-
K)/2.[(2-K)/2]^"; sl3gij(x, i, j) & "}} . "
            End If
        Next j
    Next i
49 Next i
    For i = 1 To sl4g(x)
    Print #2, "    {1/2 + (1-K)/2.[";
    For j = 1 To sl4gi(x, i)
        If j = 1 And sl4gij(x, i, j) <> 0 Then Print #2, "{1/2 + (1-K)/2.[";

```

```

        If j = 1 And s14gij(x, i, j) = 0 Then Print #2, "[ (2-K)/2]    ."
        If j > 1 And s14gij(x, i, j) <> 0 Then Print #2, "                                {1/2
+ (1-K)/2.";
        If j > 1 And s24gij(x, i, j) = 0 Then Print #2, "                                [(2-
K)/2]    ."
        If j = s14gi(x, i) And j <> 1 And s14gij(x, i, j) = 0 And x = nac And flag(x)
= 0 Then Print #2, "                                [(2-K)/2]]    ."
        If j = s14gi(x, i) And j <> 1 And s14gij(x, i, j) = 0 And x < nac Then Print
#2, "                                [(2-K)/2)]    ."
        For l = 1 To s14gij(x, i, j)
            If x = nac And flag(x) = 0 Then
                If s14gi(x, i) = 1 And s14gij(x, i, j) = 1 Then Print #2, "{1/2 + (1-
K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}}": GoTo 59
                If l = 1 And l <> s14gij(x, i, j) Then Print #2, "{1/2 + (1-K)/2. [(2-
K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l = 1 And l = s14gij(x, i, j) And j <> s14gi(x, i) Then Print #2, "{1/2
+ (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l > 1 And l <> s14gij(x, i, j) Then Print #2, "
{1/2 + (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l = s14gij(x, i, j) And l <> 1 And j <> s14gi(x, i) Then Print #2, "
{1/2 + (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l = s14gij(x, i, j) And l = 1 And j = s14gi(x, i) Then Print #2, "{1/2
+ (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l = s14gij(x, i, j) And l <> 1 And j = s14gi(x, i) Then Print #2, "
{1/2 + (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
            Else
                If s14gi(x, i) = 1 And s14gij(x, i, j) = 1 Then Print #2, "{1/2 + (1-
K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}}".": GoTo 59
                If l = 1 And l <> s14gij(x, i, j) Then Print #2, "{1/2 + (1-K)/2. [(2-
K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l = 1 And l = s14gij(x, i, j) And j <> s14gi(x, i) Then Print #2, "{1/2
+ (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l > 1 And l <> s14gij(x, i, j) Then Print #2, "
{1/2 + (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l = s14gij(x, i, j) And l <> 1 And j <> s14gi(x, i) Then Print #2, "
{1/2 + (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l = s14gij(x, i, j) And l = 1 And j = s14gi(x, i) Then Print #2, "{1/2
+ (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
                If l = s14gij(x, i, j) And l <> 1 And j = s14gi(x, i) Then Print #2, "
{1/2 + (1-K)/2. [(2-K)/2]^"; s14gijl(x, i, j, l) & "}" . "
            End If
        Next l
    Next j
59 Next i
    For i = 1 To s15g(x)
        Print #2, "                                {1/2 + (1-K)/2.";
        For j = 1 To s15gi(x, i)
            If j = 1 And s15gij(x, i, j) <> 0 Then Print #2, "{1/2 + (1-K)/2.";
            If j = 1 And s15gij(x, i, j) = 0 Then Print #2, "[ (2-K)/2]    ."
            If j > 1 And s15gij(x, i, j) <> 0 Then Print #2, "                                {1/2
+ (1-K)/2.";
            If j > 1 And j <> s15gi(x, i) And s15gij(x, i, j) = 0 Then Print #2, "
[(2-K)/2]    ."
            If j = s15gi(x, i) And j <> 1 And s15gij(x, i, j) = 0 And x = nac And flag(x)
= 0 Then Print #2, "                                [(2-K)/2]]    ."
            If j = s15gi(x, i) And j <> 1 And s15gij(x, i, j) = 0 And x < nac Then Print
#2, "                                [(2-K)/2)]    ."
            For l = 1 To s15gij(x, i, j)
                If l = 1 And s15gijl(x, i, j, l) <> 0 Then Print #2, "{1/2 + (1-K)/2.";
                If l = 1 And l <> s15gij(x, i, j) And s15gijl(x, i, j, l) = 0 Then Print #2,
"[(2-K)/2]    ."
                If l > 1 And s15gijl(x, i, j, l) <> 0 Then Print #2, "
{1/2 + (1-K)/2.";
                If l > 1 And l <> s15gij(x, i, j) And s15gijl(x, i, j, l) = 0 Then Print #2,
"                                [(2-K)/2]    ."
                If l > 1 And l = s15gij(x, i, j) And s15gijl(x, i, j, l) = 0 And j <>
s15gi(x, i) Then Print #2, "                                [(2-K)/2]]    ."
                If l > 1 And l = s15gij(x, i, j) And s15gijl(x, i, j, l) = 0 And j =
s15gi(x, i) And x = nac And flag(x) = 0 Then Print #2, "
[(2-K)/2]]    ."
                If l > 1 And l = s15gij(x, i, j) And s15gijl(x, i, j, l) = 0 And j =
s15gi(x, i) And x < nac Then Print #2, "                                [(2-
K)/2]]    ."
            For m = 1 To s15gijl(x, i, j, l)
                If x = nac And flag(x) = 0 Then
                    If s15gi(x, i) = 1 And s15gij(x, i, j) = 1 And s15gijl(x, i, j, l) = 1
Then Print #2, "{1/2 + (1-K)/2. [(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}}": GoTo 69

```

```

        If l = 1 And l <> s15gij(x, i, j) And m = 1 And m <> s15gijl(x, i, j, l)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = 1 And l <> s15gij(x, i, j) And m > 1 And m <> s15gijl(x, i, j, l)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = 1 And l <> s15gij(x, i, j) And m = s15gijl(x, i, j, l) Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l > 1 And l <> s15gij(x, i, j) And m = 1 And m <> s15gijl(x, i, j, l)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l > 1 And l <> s15gij(x, i, j) And m > 1 And m <> s15gijl(x, i, j, l)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l > 1 And l <> s15gij(x, i, j) And m = s15gijl(x, i, j, l) Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = s15gij(x, i, j) And m = 1 And m <> s15gijl(x, i, j, l) Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = s15gij(x, i, j) And m > 1 And m <> s15gijl(x, i, j, l) Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = s15gij(x, i, j) And m = s15gijl(x, i, j, l) And j <> s15gi(x, i)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = s15gij(x, i, j) And m = s15gijl(x, i, j, l) And j = s15gi(x, i)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        Else
        If s15gi(x, i) = 1 And s15gij(x, i, j) = 1 And s15gijl(x, i, j, l) = 1
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . " : GoTo
69
        If l = 1 And l <> s15gij(x, i, j) And m = 1 And m <> s15gijl(x, i, j, l)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = 1 And l <> s15gij(x, i, j) And m > 1 And m <> s15gijl(x, i, j, l)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = 1 And l <> s15gij(x, i, j) And m = s15gijl(x, i, j, l) Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l > 1 And l <> s15gij(x, i, j) And m = 1 And m <> s15gijl(x, i, j, l)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l > 1 And l <> s15gij(x, i, j) And m > 1 And m <> s15gijl(x, i, j, l)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l > 1 And l <> s15gij(x, i, j) And m = s15gijl(x, i, j, l) Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = s15gij(x, i, j) And m = 1 And m <> s15gijl(x, i, j, l) Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = s15gij(x, i, j) And m > 1 And m <> s15gijl(x, i, j, l) Then Print
#2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = s15gij(x, i, j) And m = s15gijl(x, i, j, l) And j <> s15gi(x, i)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        If l = s15gij(x, i, j) And m = s15gijl(x, i, j, l) And j = s15gi(x, i)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s15gijlm(x, i, j, l, m) & "}" . "
        End If
    Next m
Next l
Next j
69 Next i
    If x = nac And flag(x) = 0 Then
        If slgem1(x) <> 0 Then
            If slgem2(x) <> 0 And slgem3(x) <> 0 Then Print #2, " (3-3K+K^2)^";
slgem1(x) & " " . [K*(3-2K)]^"; slgem2(x) & " " . (K^2)^"; slgem3(x) & " "
            If slgem2(x) = 0 And slgem3(x) = 0 Then Print #2, " (3-3K+K^2)^";
slgem1(x) & " "
            If slgem2(x) <> 0 And slgem3(x) = 0 Then Print #2, " (3-3K+K^2)^";
slgem1(x) & " " . [K*(3-2K)]^"; slgem2(x) & " "
            If slgem2(x) = 0 And slgem3(x) <> 0 Then Print #2, " (3-3K+K^2)^";
slgem1(x) & " " . (K^2)^"; slgem3(x) & " "
            ElseIf slgem1(x) = 0 And slgem2(x) <> 0 Then
                If slgem3(x) = 0 Then Print #2, " [K*(3-2K)]^"; slgem2(x) & " "
                If slgem3(x) <> 0 Then Print #2, " [K*(3-2K)]^"; slgem2(x) & " "
(K^2)^"; slgem3(x) & " "

```

```

ElseIf slgem1(x) = 0 And slgem2(x) = 0 And slgem3(x) <> 0 Then Print #2, "
(K^2)^"; slgem3(x) & " "
End If
Else:
If slgem1(x) <> 0 Then
If slgem2(x) <> 0 And slgem3(x) <> 0 Then Print #2, " (3-3K+K^2)^";
slgem1(x) & " . [K*(3-2K)]^"; slgem2(x) & " . (K^2)^"; slgem3(x) & " . "
If slgem2(x) = 0 And slgem3(x) = 0 Then Print #2, " (3-3K+K^2)^";
slgem1(x) & " . "
If slgem2(x) <> 0 And slgem3(x) = 0 Then Print #2, " (3-3K+K^2)^";
slgem1(x) & " . [K*(3-2K)]^"; slgem2(x) & " . "
If slgem2(x) = 0 And slgem3(x) <> 0 Then Print #2, " (3-3K+K^2)^";
slgem1(x) & " . (K^2)^"; slgem3(x) & " . "
ElseIf slgem1(x) = 0 And slgem2(x) <> 0 Then
If slgem3(x) = 0 Then Print #2, " [K*(3-2K)]^"; slgem2(x) & " . "
If slgem3(x) <> 0 Then Print #2, " [K*(3-2K)]^"; slgem2(x) & " .
(K^2)^"; slgem3(x) & " . "
ElseIf slgem1(x) = 0 And slgem2(x) = 0 And slgem3(x) <> 0 Then Print #2, "
(K^2)^"; slgem3(x) & " . "
End If
End If
If flag(x) = 0 Then GoTo 100
Print #2, " Configuration 2:"
If s2pen(x) <> 0 Or s2het(x) <> 0 Or s2nsf(x) <> 0 Then
If x = 1 And nac = 2 Or s22g(x) <> 0 Or s23g(x) <> 0 Or s24g(x) <> 0 Or s25g(x)
<> 0 Or s2gem1(x) <> 0 Or s2gem2(x) <> 0 Or s2gem3(x) <> 0 Then
Print #2, " K^"; s2pen(x) & " . (1-K)^"; s2het(x) & " . (2-K)^";
s2nsf(x) & " ."
Else: Print #2, " K^"; s2pen(x) & " . (1-K)^"; s2het(x) & " . (2-K)^";
s2nsf(x) & " "
End If
End If
For i = 1 To s22g(x)
If x = 1 And nac = 2 Or i < s22g(x) Or s23g(x) <> 0 Or s24g(x) <> 0 Or s25g(x)
<> 0 Or s2gem1(x) <> 0 Or s2gem2(x) <> 0 Or s2gem3(x) <> 0 Then
Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s22gi(x, i) & " } ."
Else:
If i < s22g(x) Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s22gi(x,
i) & " } ."
If i = s22g(x) Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s22gi(x,
i) & " }"
End If
Next i
For i = 1 To s23g(x)
Print #2, " {1/2 + (1-K)/2.";
For j = 1 To s23gi(x, i)
If x = 1 And nac = 2 Or i < s23g(x) Or s24g(x) <> 0 Or s25g(x) <> 0 Or
s2gem1(x) <> 0 Or s2gem2(x) <> 0 Or s2gem3(x) <> 0 Then
If s23gi(x, i) = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s23gij(x, i,
j) & " } .": GoTo 79
If j = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s23gij(x, i, j) & " }
."
If j > 1 And j < s23gi(x, i) Then Print #2, " {1/2 +
(1-K)/2.[(2-K)/2]^"; s23gij(x, i, j) & " } ."
If j = s23gi(x, i) Then Print #2, " {1/2 + (1-
K)/2.[(2-K)/2]^"; s23gij(x, i, j) & " } } ."
Else
If s23gi(x, i) = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s23gij(x, i,
j) & " }": GoTo 79
If j = 1 Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s23gij(x, i, j) & " }
."
If j > 1 And j < s23gi(x, i) Then Print #2, " {1/2 +
(1-K)/2.[(2-K)/2]^"; s23gij(x, i, j) & " } ."
If j = s23gi(x, i) Then Print #2, " {1/2 + (1-
K)/2.[(2-K)/2]^"; s23gij(x, i, j) & " } }"
End If
Next j
79 Next i
For i = 1 To s24g(x)
Print #2, " {1/2 + (1-K)/2.";
For j = 1 To s24gi(x, i)
If j = 1 And s24gij(x, i, j) <> 0 Then Print #2, "{1/2 + (1-K)/2.";
If j = 1 And s24gij(x, i, j) = 0 Then Print #2, "[(2-K)/2] ."
If j > 1 And s24gij(x, i, j) <> 0 Then Print #2, " {1/2
+ (1-K)/2.";
If j > 1 And s24gij(x, i, j) = 0 Then Print #2, " [(2-
K)/2] ."

```

```

      If j = s24gi(x, i) And j <> 1 And s24gij(x, i, j) = 0 And x = nac Then Print
#2, " [(2-K)/2]] "
      If j = s24gi(x, i) And j <> 1 And s24gij(x, i, j) = 0 And x < nac Then Print
#2, " [(2-K)/2]] ."
      For l = 1 To s24gij(x, i, j)
        If x = 1 And nac = 2 Or i < s24g(x) Or s25g(x) <> 0 Or s2gem1(x) <> 0 Or
s2gem2(x) <> 0 Or s2gem3(x) <> 0 Then
          If s24gi(x, i) = 1 And s24gij(x, i, j) = 1 Then Print #2, "{1/2 + (1-
K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} . ": GoTo 89
          If l = 1 And l <> s24gij(x, i, j) Then Print #2, "{1/2 + (1-K)/2.[(2-
K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l = 1 And l = s24gij(x, i, j) And j <> s24gi(x, i) Then Print #2, "{1/2
+ (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l > 1 And l <> s24gij(x, i, j) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l = s24gij(x, i, j) And l <> 1 And j <> s24gi(x, i) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l = s24gij(x, i, j) And l = 1 And j = s24gi(x, i) Then Print #2, "{1/2
+ (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l = s24gij(x, i, j) And l <> 1 And j = s24gi(x, i) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
        Else:
          If s24gi(x, i) = 1 And s24gij(x, i, j) = 1 Then Print #2, "{1/2 + (1-
K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}}": GoTo 89
          If l = 1 And l <> s24gij(x, i, j) Then Print #2, "{1/2 + (1-K)/2.[(2-
K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l = 1 And l = s24gij(x, i, j) And j <> s24gi(x, i) Then Print #2, "{1/2
+ (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l > 1 And l <> s24gij(x, i, j) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l = s24gij(x, i, j) And l <> 1 And j <> s24gi(x, i) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l = s24gij(x, i, j) And l = 1 And j = s24gi(x, i) Then Print #2, "{1/2
+ (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
          If l = s24gij(x, i, j) And l <> 1 And j = s24gi(x, i) Then Print #2, "
{1/2 + (1-K)/2.[(2-K)/2]^"; s24gijl(x, i, j, 1) & "}} ."
        End If
      Next l
    Next j
  89 Next i
  For i = 1 To s25g(x)
    Print #2, " {1/2 + (1-K)/2.";
    For j = 1 To s25gi(x, i)
      If j = 1 And s25gij(x, i, j) <> 0 Then Print #2, "{1/2 + (1-K)/2.";
      If j = 1 And s25gij(x, i, j) = 0 Then Print #2, "[(2-K)/2] ."
      If j > 1 And s25gij(x, i, j) <> 0 Then Print #2, " {1/2
+ (1-K)/2.";
      If j > 1 And j <> s25gi(x, i) And s25gij(x, i, j) = 0 Then Print #2, "
[(2-K)/2] ."
      If j = s25gi(x, i) And j <> 1 And s25gij(x, i, j) = 0 And x = nac Then Print
#2, " [(2-K)/2]] "
      If j = s25gi(x, i) And j <> 1 And s25gij(x, i, j) = 0 And x < nac Then Print
#2, " [(2-K)/2]] ."
      For l = 1 To s25gij(x, i, j)
        If l = 1 And s25gijl(x, i, j, 1) <> 0 Then Print #2, "{1/2 + (1-K)/2.";
        If l = 1 And l <> s25gij(x, i, j) And s25gijl(x, i, j, 1) = 0 Then Print #2,
"[(2-K)/2] ."
        If l > 1 And s25gijl(x, i, j, 1) <> 0 Then Print #2, "
{1/2 + (1-K)/2.";
        If l > 1 And l <> s25gij(x, i, j) And s25gijl(x, i, j, 1) = 0 Then Print #2,
" [(2-K)/2] ."
        If l > 1 And l = s25gij(x, i, j) And s25gijl(x, i, j, 1) = 0 And j <>
s25gi(x, i) Then Print #2, " [(2-K)/2]] ."
        If l > 1 And l = s25gij(x, i, j) And s25gijl(x, i, j, 1) = 0 And j =
s25gi(x, i) And x = nac Then Print #2, " [(2-
K)/2]] "
        If l > 1 And l = s25gij(x, i, j) And s25gijl(x, i, j, 1) = 0 And j =
s25gi(x, i) And x < nac Then Print #2, " [(2-
K)/2]] ."
      For m = 1 To s25gijl(x, i, j, 1)
        If x = 1 And nac = 2 Or i < s25g(x) Or s2gem1(x) <> 0 Or s2gem2(x) <> 0 Or
s2gem3(x) <> 0 Then
          If s25gi(x, i) = 1 And s25gij(x, i, j) = 1 And s25gijl(x, i, j, 1) = 1
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, 1, m) & "}} . ": GoTo
99
          If l = 1 And l <> s25gij(x, i, j) And m = 1 And m <> s25gijl(x, i, j, 1)
Then Print #2, "{1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, 1, m) & "}} ."

```

```

        If l = 1 And l <> s25gij(x, i, j) And m > 1 And m <> s25gijl(x, i, j, l)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l = 1 And l <> s25gij(x, i, j) And m = s25gijl(x, i, j, l) Then Print
#2, " {1/2 + (1-K)/2.[(2-K)/2]^";
s25gijlm(x, i, j, l, m) & " } } . "
        If l > 1 And l <> s25gij(x, i, j) And m = 1 And m <> s25gijl(x, i, j, l)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l > 1 And l <> s25gij(x, i, j) And m > 1 And m <> s25gijl(x, i, j, l)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l > 1 And l <> s25gij(x, i, j) And m = s25gijl(x, i, j, l) Then Print
#2, " {1/2 + (1-K)/2.[(2-K)/2]^";
s25gijlm(x, i, j, l, m) & " } } . "
        If l = s25gij(x, i, j) And m = 1 And m <> s25gijl(x, i, j, l) Then Print
#2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l = s25gij(x, i, j) And m > 1 And m <> s25gijl(x, i, j, l) Then Print
#2, " {1/2 + (1-K)/2.[(2-K)/2]^";
s25gijlm(x, i, j, l, m) & " } . "
        If l = s25gij(x, i, j) And m = s25gijl(x, i, j, l) And j <> s25gi(x, i)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } } } . "
        If l = s25gij(x, i, j) And m = s25gijl(x, i, j, l) And j = s25gi(x, i)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } } } } . "
Else:
        If s25gi(x, i) = 1 And s25gij(x, i, j) = 1 And s25gijl(x, i, j, l) = 1
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } } } } : GoTo 99
        If l = 1 And l <> s25gij(x, i, j) And m = 1 And m <> s25gijl(x, i, j, l)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l = 1 And l <> s25gij(x, i, j) And m > 1 And m <> s25gijl(x, i, j, l)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l = 1 And l <> s25gij(x, i, j) And m = s25gijl(x, i, j, l) Then Print
#2, " {1/2 + (1-K)/2.[(2-K)/2]^";
s25gijlm(x, i, j, l, m) & " } } . "
        If l > 1 And l <> s25gij(x, i, j) And m = 1 And m <> s25gijl(x, i, j, l)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l > 1 And l <> s25gij(x, i, j) And m > 1 And m <> s25gijl(x, i, j, l)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l > 1 And l <> s25gij(x, i, j) And m = s25gijl(x, i, j, l) Then Print
#2, " {1/2 + (1-K)/2.[(2-K)/2]^";
s25gijlm(x, i, j, l, m) & " } } . "
        If l = s25gij(x, i, j) And m = 1 And m <> s25gijl(x, i, j, l) Then Print
#2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } . "
        If l = s25gij(x, i, j) And m > 1 And m <> s25gijl(x, i, j, l) Then Print
#2, " {1/2 + (1-K)/2.[(2-K)/2]^";
s25gijlm(x, i, j, l, m) & " } . "
        If l = s25gij(x, i, j) And m = s25gijl(x, i, j, l) And j <> s25gi(x, i)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } } } . "
        If l = s25gij(x, i, j) And m = s25gijl(x, i, j, l) And j = s25gi(x, i)
Then Print #2, " {1/2 + (1-K)/2.[(2-K)/2]^"; s25gijlm(x, i, j, l, m) & " } } } } . "
        End If
    Next m
Next l
Next j
99 Next i
    If x = 1 And nac = 2 Or s2gem1(x) <> 0 Or s2gem2(x) <> 0 Or s2gem3(x) <> 0 Then
        If s2gem1(x) <> 0 Then
            If s2gem2(x) <> 0 And s2gem3(x) <> 0 Then Print #2, " (3-3K+K^2)^";
s2gem1(x) & " . [K*(3-2K)]^"; s2gem2(x) & " . (K^2)^"; s2gem3(x) & " . "
            If s2gem2(x) = 0 And s2gem3(x) = 0 Then Print #2, " (3-3K+K^2)^"; s2gem1(x)
& " . "
            If s2gem2(x) <> 0 And s2gem3(x) = 0 Then Print #2, " (3-3K+K^2)^";
s2gem1(x) & " . [K*(3-2K)]^"; s2gem2(x) & " . "
            If s2gem2(x) = 0 And s2gem3(x) <> 0 Then Print #2, " (3-3K+K^2)^";
s2gem1(x) & " . (K^2)^"; s2gem3(x) & " . "
            ElseIf s2gem1(x) = 0 And s2gem2(x) <> 0 Then
                If s2gem3(x) = 0 Then Print #2, " [K*(3-2K)]^"; s2gem2(x) & " . "
                If s2gem3(x) <> 0 Then Print #2, " [K*(3-2K)]^"; s2gem2(x) & " . (K^2)^";
s2gem3(x) & " . "
            ElseIf s2gem1(x) = 0 And s2gem2(x) = 0 And s2gem3(x) <> 0 Then Print #2, "
(K^2)^"; s2gem3(x) & " . "
        End If

```

```

Else:
  If s2gem1(x) <> 0 Then
    If s2gem2(x) <> 0 And s2gem3(x) <> 0 Then Print #2, "      (3-3K+K^2)^";
s2gem1(x) & " . [K*(3-2K)]^"; s2gem2(x) & " . (K^2)^"; s2gem3(x) & " "
    If s2gem2(x) = 0 And s2gem3(x) = 0 Then Print #2, "      (3-3K+K^2)^"; s2gem1(x)
& ""
    If s2gem2(x) <> 0 And s2gem3(x) = 0 Then Print #2, "      (3-3K+K^2)^";
s2gem1(x) & " . [K*(3-2K)]^"; s2gem2(x) & ""
    If s2gem2(x) = 0 And s2gem3(x) <> 0 Then Print #2, "      (3-3K+K^2)^";
s2gem1(x) & " . (K^2)^"; s2gem3(x) & ""
    ElseIf s2gem1(x) = 0 And s2gem2(x) <> 0 Then
      If s2gem3(x) = 0 Then Print #2, "      [K*(3-2K)]^"; s2gem2(x) & ""
      If s2gem3(x) <> 0 Then Print #2, "      [K*(3-2K)]^"; s2gem2(x) & " . (K^2)^";
s2gem3(x) & ""
    ElseIf s2gem1(x) = 0 And s2gem2(x) = 0 And s2gem3(x) <> 0 Then Print #2, "
(K^2)^"; s2gem3(x) & ""
    End If
  End If
100 Next x
Close #2
Open App.Path & "\function.txt" For Input As #2
Do While Not EOF(2)
  frm_inicial.txt_function.Text = Input$(LOF(2), 2)
Loop
Close #2
frm_inicial.Label5.Visible = True: frm_inicial.txt_function.Visible = True
frm_inicial.Refresh
End Sub

```

Public Sub Confint()

```

  frm_inicial.Label3.Visible = True: frm_inicial.Refresh
  marc = 0
10  t = 1: k0 = 0: kn = 1
  h = (kn - k0) / 1000: k = k0: GoSub functsub: s = pk
  k = kn: GoSub functsub: s = s + pk: w = 4
  For j1 = 1 To 999
    k = k0 + j1 * h
    GoSub functsub
    s = s + w * pk
    w = 6 - w
  Next j1
  totalarea = h * s / 3: s = 0
  k0 = 0.001: niter = 100
repeat:
  k = k0: GoSub functsub: pk0 = pk: k1 = 1
  For j1 = 1 To niter
    k1 = k1 - 1 / niter: k = k1: GoSub functsub
    pk1 = pk: If pk1 >= pk0 Then GoTo getout
  Next j1
getout:  k0 = k0: kn = k1: GoSub simpsonsrule
  If area / totalarea >= 0.95 Then k0 = k0 + 1 / niter: GoTo repeat
  k0 = k0 - 1 / niter: k1 = k1 + 1 / niter: niter = niter * 2
  GoTo repeat
simpsonsrule:
  h = (kn - k0) / 1000: k = k0: GoSub functsub: s = pk
  k = kn: GoSub functsub: s = s + pk: w = 4
  For j1 = 1 To 999
    k = k0 + j1 * h
    GoSub functsub
    s = s + w * pk
    w = 6 - w
  Next j1
  area = h * s / 3: s = 0
  If t = 1 Then k01 = k0: kn1 = kn: t = t + 1: GoTo ifabs
  If t = 2 Then k02 = k0: kn2 = kn: t = t - 1
  dif0 = Abs(0.95 - area / totalarea): dif1 = Abs(k01 - k02): dif2 = Abs(kn1 - kn2)
  frm_inicial.Refresh
  If dif0 <= 0.0001 And dif1 <= 0.001 And dif2 <= 0.001 Then
20  If nac <> 0 Then
    If marc = 0 Then 'duas S1
      lik1 = k0: lsk1 = kn: marc = 1: GoTo 10
    ElseIf marc = 1 Then 'duas S2
      lik2 = k0: lsk2 = kn
      If flag(1) = 0 Or flag(2) = 0 Or nac = 1 Then
        lifinal = (lik1 / varkc1 + lik2 / varkc2) / (1 / varkc1 + 1 / varkc2)
        lsfinal = (lsk1 / varkc1 + lsk2 / varkc2) / (1 / varkc1 + 1 / varkc2)

```



```

        frm_inicial.Label3.Visible = False: frm_inicial.Refresh
        frm_inicial.txt_conf1.Text = Format$(lifinal, "0.000")
        frm_inicial.txt_conf2.Text = Format$(lsfinal, "0.000")
        GoTo endprogram
    Else: lik2 = k0: lsk2 = kn: marc = 2: GoTo 10
    End If
    ElseIf marc = 2 Then
        lik3 = k0: lsk3 = kn: marc = 3: GoTo 10
    ElseIf marc = 3 Then
        lik4 = k0: lsk4 = kn
        lifinal = (lik1 / varkc1 + lik2 / varkc2 + lik3 / varkc3 + lik4 / varkc4)
        lifinal = lifinal / (1 / varkc1 + 1 / varkc2 + 1 / varkc3 + 1 / varkc4)
        lsfinal = (lsk1 / varkc1 + lsk2 / varkc2 + lsk3 / varkc3 + lsk4 / varkc4)
        lsfinal = lsfinal / (1 / varkc1 + 1 / varkc2 + 1 / varkc3 + 1 / varkc4)
        frm_inicial.Label3.Visible = False: frm_inicial.Refresh
        frm_inicial.txt_conf1.Text = Format$(lifinal, "0.000")
        frm_inicial.txt_conf2.Text = Format$(lsfinal, "0.000")
        GoTo endprogram
    End If
    Else: frm_inicial.Label3.Visible = False: frm_inicial.Refresh
        frm_inicial.txt_conf1.Text = Format$(k0, "0.000"): frm_inicial.txt_conf2.Text =
Format$(kn, "0.000")
        GoTo endprogram
    End If
    End If
    ifabs:
        If Abs(0.95 - area / totalarea) <= 0.0001 Then GoTo 20
        k = k0: GoSub functsub: pk0 = pk
        k = kn: GoSub functsub: pk1 = pk
    Return
    functsub:
        pk = k ^ pen * (1 - k) ^ hetnpen * (2 - k) ^ norsf
        For i = 1 To n2g
            pk = pk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ n2gi(i))
        Next i
        For i = 1 To n3g
            plk = 1
            For j = 1 To n3gi(i)
                plk = plk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ n3gij(i, j))
            Next j
            plk = plk * (1 - k) / 2 + 1 / 2: pk = pk * plk
        Next i
        For i = 1 To n4g
            p2k = 1: p3k = 1
            For j = 1 To n4gi(i)
                For l = 1 To n4gij(i, j)
                    p2k = p2k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ n4gijl(i, j, l))
                Next l
                p3k = p3k * (p2k * (1 - k) / 2 + 1 / 2): p2k = 1
            Next j
            p3k = p3k * (1 - k) / 2 + 1 / 2: pk = pk * p3k
        Next i
        For i = 1 To n5g
            p4k = 1: p5k = 1: p6k = 1
            For j = 1 To n5gi(i)
                For l = 1 To n5gij(i, j)
                    For m = 1 To n5gijl(i, j, l)
                        p5k = p5k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ n5gijlm(i, j, l, m))
                    Next m
                    p6k = p6k * (p5k * (1 - k) / 2 + 1 / 2): p5k = 1
                Next l
                p4k = p4k * (p6k * (1 - k) / 2 + 1 / 2): p6k = 1
            Next j
            p4k = p4k * (1 - k) / 2 + 1 / 2: pk = pk * p4k
        Next i
        For x = 1 To nac
            If marc = 0 Then
                pk = pk * k ^ slpen(x) * (1 - k) ^ slhet(x) * (2 - k) ^ slnsf(x)
                For i = 1 To sl2g(x): pk = pk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl2gi(x,
i)): Next i
                For i = 1 To sl3g(x)
                    plk = 1
                    For j = 1 To sl3gi(x, i): plk = plk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
sl3gij(x, i, j)): Next j
                    plk = plk * (1 - k) / 2 + 1 / 2: pk = pk * plk
                Next i
                For i = 1 To sl4g(x)

```

```

p2k = 1: p3k = 1
For j = 1 To s14gi(x, i)
  For l = 1 To s14gij(x, i, j): p2k = p2k * (1 / 2 + (1 - k) / 2 * ((2 - k) /
2) ^ s14gijl(x, i, j, l)): Next l
  p3k = p3k * (p2k * (1 - k) / 2 + 1 / 2): p2k = 1
Next j
p3k = p3k * (1 - k) / 2 + 1 / 2: pk = pk * p3k
Next i
For i = 1 To s15g(x)
  p4k = 1: p5k = 1: p6k = 1
  For j = 1 To s15gi(x, i)
    For l = 1 To s15gij(x, i, j)
      For m = 1 To s15gijl(x, i, j, l)
        p5k = p5k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s15gijlm(x, i, j, l,
m))
        Next m
        p6k = p6k * (p5k * (1 - k) / 2 + 1 / 2): p5k = 1
        Next l
        p4k = p4k * (p6k * (1 - k) / 2 + 1 / 2): p6k = 1
        Next j
        p4k = p4k * (1 - k) / 2 + 1 / 2: pk = pk * p4k
        Next i
        For i = 1 To slgem1(x): pk = pk * ((3 - 3 * k + k ^ 2) / 3): Next i
        For i = 1 To slgem2(x): pk = pk * (k * (3 - 2 * k) / 6): Next i
        For i = 1 To slgem3(x): pk = pk * (k ^ 2 / 3): Next i
      ElseIf marc = 1 Then
        pk = pk * k ^ s2pen(x) * (1 - k) ^ s2het(x) * (2 - k) ^ s2nsf(x)
        For i = 1 To s22g(x): pk = pk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s22gi(x,
i)): Next i
        For i = 1 To s23g(x)
          p1k = 1
          For j = 1 To s23gi(x, i): p1k = p1k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
s23gij(x, i, j)): Next j
          p1k = p1k * (1 - k) / 2 + 1 / 2: pk = pk * p1k
          Next i
          For i = 1 To s24g(x)
            p2k = 1: p3k = 1
            For j = 1 To s24gi(x, i)
              For l = 1 To s24gij(x, i, j): p2k = p2k * (1 / 2 + (1 - k) / 2 * ((2 - k) /
2) ^ s24gijl(x, i, j, l)): Next l
              p3k = p3k * (p2k * (1 - k) / 2 + 1 / 2): p2k = 1
              Next j
              p3k = p3k * (1 - k) / 2 + 1 / 2: pk = pk * p3k
              Next i
              For i = 1 To s25g(x)
                p4k = 1: p5k = 1: p6k = 1
                For j = 1 To s25gi(x, i)
                  For l = 1 To s25gij(x, i, j)
                    For m = 1 To s25gijl(x, i, j, l)
                      p5k = p5k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s25gijlm(x, i, j, l,
m))
                      Next m
                      p6k = p6k * (p5k * (1 - k) / 2 + 1 / 2): p5k = 1
                      Next l
                      p4k = p4k * (p6k * (1 - k) / 2 + 1 / 2): p6k = 1
                      Next j
                      p4k = p4k * (1 - k) / 2 + 1 / 2: pk = pk * p4k
                      Next i
                      For i = 1 To s2gem1(x): pk = pk * ((3 - 3 * k + k ^ 2) / 3): Next i
                      For i = 1 To s2gem2(x): pk = pk * (k * (3 - 2 * k) / 6): Next i
                      For i = 1 To s2gem3(x): pk = pk * (k ^ 2 / 3): Next i
                    ElseIf marc = 2 Then
                      If x = 1 Then
                        pk = pk * k ^ slpen(x) * (1 - k) ^ slhet(x) * (2 - k) ^ slnsf(x)
                        For i = 1 To sl2g(x): pk = pk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
sl2gi(x, i)): Next i
                        For i = 1 To sl3g(x)
                          p1k = 1
                          For j = 1 To sl3gi(x, i): p1k = p1k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
sl3gij(x, i, j)): Next j
                          p1k = p1k * (1 - k) / 2 + 1 / 2: pk = pk * p1k
                          Next i
                          For i = 1 To sl4g(x)
                            p2k = 1: p3k = 1
                            For j = 1 To sl4gi(x, i)
                              For l = 1 To sl4gij(x, i, j): p2k = p2k * (1 / 2 + (1 - k) / 2 * ((2 - k)
/ 2) ^ sl4gijl(x, i, j, l)): Next l

```

```

        p3k = p3k * (p2k * (1 - k) / 2 + 1 / 2): p2k = 1
    Next j
    p3k = p3k * (1 - k) / 2 + 1 / 2: pk = pk * p3k
Next i
For i = 1 To s15g(x)
    p4k = 1: p5k = 1: p6k = 1
    For j = 1 To s15gi(x, i)
        For l = 1 To s15gij(x, i, j)
            For m = 1 To s15gijl(x, i, j, l)
                p5k = p5k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s15gijlm(x, i, j,
1, m))
                Next m
                p6k = p6k * (p5k * (1 - k) / 2 + 1 / 2): p5k = 1
            Next l
            p4k = p4k * (p6k * (1 - k) / 2 + 1 / 2): p6k = 1
        Next j
        p4k = p4k * (1 - k) / 2 + 1 / 2: pk = pk * p4k
    Next i
    For i = 1 To s1gem1(x): pk = pk * ((3 - 3 * k + k ^ 2) / 3): Next i
    For i = 1 To s1gem2(x): pk = pk * (k * (3 - 2 * k) / 6): Next i
    For i = 1 To s1gem3(x): pk = pk * (k ^ 2 / 3): Next i
ElseIf x = 2 Then
    pk = pk * k ^ s2pen(x) * (1 - k) ^ s2het(x) * (2 - k) ^ s2nsf(x)
    For i = 1 To s22g(x): pk = pk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
s22gi(x, i)): Next i
    For i = 1 To s23g(x)
        p1k = 1
        For j = 1 To s23gi(x, i): p1k = p1k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
s23gij(x, i, j)): Next j
        p1k = p1k * (1 - k) / 2 + 1 / 2: pk = pk * p1k
    Next i
    For i = 1 To s24g(x)
        p2k = 1: p3k = 1
        For j = 1 To s24gi(x, i)
            For l = 1 To s24gij(x, i, j): p2k = p2k * (1 / 2 + (1 - k) / 2 * ((2 - k)
/ 2) ^ s24gijl(x, i, j, l)): Next l
            p3k = p3k * (p2k * (1 - k) / 2 + 1 / 2): p2k = 1
        Next j
        p3k = p3k * (1 - k) / 2 + 1 / 2: pk = pk * p3k
    Next i
    For i = 1 To s25g(x)
        p4k = 1: p5k = 1: p6k = 1
        For j = 1 To s25gi(x, i)
            For l = 1 To s25gij(x, i, j)
                For m = 1 To s25gijl(x, i, j, l)
                    p5k = p5k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s25gijlm(x, i, j,
1, m))
                Next m
                p6k = p6k * (p5k * (1 - k) / 2 + 1 / 2): p5k = 1
            Next l
            p4k = p4k * (p6k * (1 - k) / 2 + 1 / 2): p6k = 1
        Next j
        p4k = p4k * (1 - k) / 2 + 1 / 2: pk = pk * p4k
    Next i
    For i = 1 To s2gem1(x): pk = pk * ((3 - 3 * k + k ^ 2) / 3): Next i
    For i = 1 To s2gem2(x): pk = pk * (k * (3 - 2 * k) / 6): Next i
    For i = 1 To s2gem3(x): pk = pk * (k ^ 2 / 3): Next i
End If
ElseIf marc = 3 Then
    If x = 1 Then
        pk = pk * k ^ s2pen(x) * (1 - k) ^ s2het(x) * (2 - k) ^ s2nsf(x)
        For i = 1 To s22g(x): pk = pk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
s22gi(x, i)): Next i
        For i = 1 To s23g(x)
            p1k = 1
            For j = 1 To s23gi(x, i): p1k = p1k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
s23gij(x, i, j)): Next j
            p1k = p1k * (1 - k) / 2 + 1 / 2: pk = pk * p1k
        Next i
        For i = 1 To s24g(x)
            p2k = 1: p3k = 1
            For j = 1 To s24gi(x, i)
                For l = 1 To s24gij(x, i, j): p2k = p2k * (1 / 2 + (1 - k) / 2 * ((2 - k)
/ 2) ^ s24gijl(x, i, j, l)): Next l
                p3k = p3k * (p2k * (1 - k) / 2 + 1 / 2): p2k = 1
            Next j
            p3k = p3k * (1 - k) / 2 + 1 / 2: pk = pk * p3k
        Next i
    End If

```

```

Next i
For i = 1 To s25g(x)
    p4k = 1: p5k = 1: p6k = 1
    For j = 1 To s25gi(x, i)
        For l = 1 To s25gij(x, i, j)
            For m = 1 To s25gijl(x, i, j, l)
                p5k = p5k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ s25gijlm(x, i, j,
1, m))
                Next m
                p6k = p6k * (p5k * (1 - k) / 2 + 1 / 2): p5k = 1
                Next l
                p4k = p4k * (p6k * (1 - k) / 2 + 1 / 2): p6k = 1
                Next j
                p4k = p4k * (1 - k) / 2 + 1 / 2: pk = pk * p4k
            Next i
            For i = 1 To s2gem1(x): pk = pk * ((3 - 3 * k + k ^ 2) / 3): Next i
            For i = 1 To s2gem2(x): pk = pk * (k * (3 - 2 * k) / 6): Next i
            For i = 1 To s2gem3(x): pk = pk * (k ^ 2 / 3): Next i
        ElseIf x = 2 Then
            pk = pk * k ^ slpen(x) * (1 - k) ^ slhet(x) * (2 - k) ^ slnsf(x)
            For i = 1 To sl2g(x): pk = pk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
s12gi(x, i)): Next i
            For i = 1 To sl3g(x)
                plk = 1
                For j = 1 To sl3gi(x, i): plk = plk * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^
s13gij(x, i, j)): Next j
                plk = plk * (1 - k) / 2 + 1 / 2: pk = pk * plk
            Next i
            For i = 1 To sl4g(x)
                p2k = 1: p3k = 1
                For j = 1 To sl4gi(x, i)
                    For l = 1 To sl4gij(x, i, j): p2k = p2k * (1 / 2 + (1 - k) / 2 * ((2 - k)
/ 2) ^ sl4gijl(x, i, j, l)): Next l
                    p3k = p3k * (p2k * (1 - k) / 2 + 1 / 2): p2k = 1
                Next j
                p3k = p3k * (1 - k) / 2 + 1 / 2: pk = pk * p3k
            Next i
            For i = 1 To sl5g(x)
                p4k = 1: p5k = 1: p6k = 1
                For j = 1 To sl5gi(x, i)
                    For l = 1 To sl5gij(x, i, j)
                        For m = 1 To sl5gijl(x, i, j, l)
                            p5k = p5k * (1 / 2 + (1 - k) / 2 * ((2 - k) / 2) ^ sl5gijlm(x, i, j,
1, m))
                            Next m
                            p6k = p6k * (p5k * (1 - k) / 2 + 1 / 2): p5k = 1
                            Next l
                            p4k = p4k * (p6k * (1 - k) / 2 + 1 / 2): p6k = 1
                            Next j
                            p4k = p4k * (1 - k) / 2 + 1 / 2: pk = pk * p4k
                        Next i
                        For i = 1 To slgem1(x): pk = pk * ((3 - 3 * k + k ^ 2) / 3): Next i
                        For i = 1 To slgem2(x): pk = pk * (k * (3 - 2 * k) / 6): Next i
                        For i = 1 To slgem3(x): pk = pk * (k ^ 2 / 3): Next i
                    End If
                End If
            Next x
            For i = 1 To ngem1: pk = pk * ((3 - 3 * k + k ^ 2) / 3): Next i
            For i = 1 To ngem2: pk = pk * (k * (3 - 2 * k) / 6): Next i
            For i = 1 To ngem3: pk = pk * (k ^ 2 / 3): Next i
        Return
    endprogram:
End Sub

```

Public Sub LimpaVar()

```

pen = Empty: frm_screen.txt_np.Text = Empty
hetnpen = Empty: frm_screen.txt_nhetnp.Text = Empty
norsf = Empty: frm_screen.txt_nsf.Text = Empty
frm_write.txt_write.Text = Empty
kone = Empty: k0 = Empty: kn = Empty: lik1 = Empty: lik2 = Empty
lik3 = Empty: lik4 = Empty: lsk1 = Empty: lsk2 = Empty: lsk3 = Empty
lsk4 = Empty: lifinal = Empty: lsfinal = Empty
kc1 = Empty: kc2 = Empty: kc3 = Empty: kc4 = Empty
varkc1 = Empty: varkc2 = Empty: varkc3 = Empty: varkc4 = Empty
n2g = Empty
For i = 1 To 15: n2gi(i) = Empty: Next i

```

```

n3g = Empty
For i = 1 To 10
    n3gi(i) = Empty
    For j = 1 To 10: n3gij(i, j) = Empty: Next j
Next i
n4g = Empty
For i = 1 To 10
    n4gi(i) = Empty
    For j = 1 To 10
        n4gij(i, j) = Empty
        For l = 1 To 10: n4gijl(i, j, l) = Empty: Next l
    Next j
Next i
n5g = Empty
For i = 1 To 10
    n5gi(i) = Empty
    For j = 1 To 10
        n5gij(i, j) = Empty
        For l = 1 To 10
            n5gijl(i, j, l) = Empty
            For m = 1 To 10: n5gijlm(i, j, l, m) = Empty: Next m
        Next l
    Next j
Next i
nac = Empty
For i = 1 To 2
    slpen(i) = Empty: slhet(i) = Empty: slnsf(i) = Empty
    s2pen(i) = Empty: s2het(i) = Empty: s2nsf(i) = Empty
    sl2g(i) = Empty: sl3g(i) = Empty: sl4g(i) = Empty: sl5g(i) = Empty
    s22g(i) = Empty: s23g(i) = Empty: s24g(i) = Empty: s25g(i) = Empty
    slgem1(i) = Empty: slgem2(i) = Empty: slgem3(i) = Empty
    s2gem1(i) = Empty: s2gem2(i) = Empty: s2gem3(i) = Empty
    For j = 1 To 15: sl2gi(i, j) = Empty: s22gi(i, j) = Empty: Next j
Next i
For i = 1 To 2
    For j = 1 To 10
        sl3gi(i, j) = Empty: s23gi(i, j) = Empty: sl4gi(i, j) = Empty
        s24gi(i, j) = Empty
        For l = 1 To 10
            sl3gij(i, j, l) = Empty: s23gij(i, j, l) = Empty
            sl4gij(i, j, l) = Empty: s24gij(i, j, l) = Empty
            For m = 1 To 10
                sl4gijl(i, j, l, m) = Empty: s24gijl(i, j, l, m) = Empty
            Next m
        Next l
    Next j
Next i
For i = 1 To 2
    For j = 1 To 10
        sl5gi(i, j) = Empty: s25gi(i, j) = Empty
        For l = 1 To 10
            sl5gij(i, j, l) = Empty: s25gij(i, j, l) = Empty
            For m = 1 To 10
                sl5gijl(i, j, l, m) = Empty: s25gijl(i, j, l, m) = Empty
                For n = 1 To 10
                    sl5gijlm(i, j, l, m, n) = Empty: s25gijlm(i, j, l, m, n) = Empty
                Next n
            Next m
        Next l
    Next j
Next i
ngem1 = Empty: ngem2 = Empty: ngem3 = Empty
frm_inicial.txt_function.Text = Empty
frm_inicial.txt_function.Visible = False
frm_inicial.Label5.Visible = False: frm_inicial.Label6.Visible = False
frm_inicial.Label11.Visible = False
frm_inicial.Label7.Visible = False: frm_inicial.txt_kvalue.Text = Empty
frm_inicial.txt_kvalue.Visible = False
frm_inicial.txt_conf1.Text = Empty: frm_inicial.txt_conf1.Visible = False
frm_inicial.txt_conf2.Text = Empty: frm_inicial.txt_conf2.Visible = False
frm_inicial.Frame1.Visible = False: frm_inicial.Label10.Visible = False
frm_inicial.txt_ident.Visible = False
frm_inicial.mnuFileSaveas.Enabled = False
frm_inicial.mnuFilePrint.Enabled = False
frm_inicial.txt_kc2.Visible = False: frm_inicial.txt_kc4.Visible = False
For i = 1 To 4: frm_inicial.Label9(i - 1).Visible = False: Next i
frm_inicial.txt_k1.Visible = False: frm_inicial.txt_k2.Visible = False

```

```

frm_inicial.Label12(0).Visible = False
frm_inicial.Label12(1).Visible = False
frm_inicial.txt_ident.Text = Empty
End Sub

```

FORMULÁRIOS

FRM_INICIAL

```

Private Sub mnudatacreate_Click()
    Call LimpaVar
    frm_write.Visible = True
End Sub

Private Sub mnudataopen_Click()
    Call LimpaVar
    On Error GoTo dbErrorHandler
    cdbOpen.Filter = "Text docs (*.txt)|*.txt": cdbOpen.DefaultExt = "txt"
    cdbOpen.ShowOpen
    Open cdbOpen.FileName For Input As #1
    Call ReadFile
    Call Run
    Call WriteFunction
    Call Confint
    frm_inicial.Label1.Visible = True: frm_inicial.Label7.Visible = True
    frm_inicial.txt_conf1.Visible = True: frm_inicial.txt_conf2.Visible = True
    GoTo endprogram
dbErrorHandler:
    Exit Sub
endprogram:
End Sub

Private Sub mnudatascreen_Click()
    Call LimpaVar
    frm_screen.Visible = True: frm_screen.txt_np.SetFocus
    frm_inicial.txt_function.Visible = False: frm_inicial.Label6.Visible = False:
    frm_inicial.Label7.Visible = False
End Sub

Private Sub mnuFileExit_Click()
    End
End Sub

Private Sub mnuFilePrint_Click()
    Dim MyFont As New StdFont
    Dim sex As String
    Dim intNumCopies, intI As Integer
    With MyFont
        .Bold = True
        .Name = "Courier New"
        .Size = 9
    End With
    Set Printer.Font = MyFont
    On Error GoTo dbErrorHandler
    cdbPrint.ShowPrinter
    intNumCopies = cdbPrint.Copies
    For intI = 1 To intNumCopies
        Printer.Print
        Printer.Print
        Printer.Print "Identification : " & frm_inicial.txt_ident.Text
        Printer.Print
        Printer.Print "Likelihood function"
        Printer.Print
        Printer.Print frm_inicial.txt_function.Text
        Printer.Print
        If nac <> 0 Then
            If kc3 = 0 And kc4 = 0 Then
                Printer.Print "K1 = " & frm_inicial.txt_k1.Text
                Printer.Print "K2 = " & frm_inicial.txt_k2.Text
            Else:
                Printer.Print "K11 = " & frm_inicial.txt_k1.Text
                Printer.Print "K12 = " & frm_inicial.txt_k2.Text
            End If
        End If
    Next intI
End Sub

```

```

        Printer.Print "K21 = " & frm_inicial.txt_kc4.Text
        Printer.Print "K22 = " & frm_inicial.txt_kc2.Text
    End If
End If
Printer.Print
Printer.Print "K = " & frm_inicial.txt_kvalue.Text
Printer.Print
Printer.Print "CI95%(K) = " & frm_inicial.txt_conf1.Text & " - " &
frm_inicial.txt_conf2.Text
Printer.NewPage
Next intI
Printer.EndDoc
Refresh
frm_inicial.Visible = True
Exit Sub
dbErrHandler:
End Sub

Private Sub mnuFileSaveas_Click()
    CdbSaveas.Flags = &H400 + &H2: CdbSaveas.Filter = "Text docs (*.txt)|*.txt"
    CdbSaveas.DefaultExt = "txt": CdbSaveas.ShowSave
    On Error GoTo dbErrHandler
    Open CdbSaveas.FileTitle For Output As #1
    Print #1,
    Print #1,
    Print #1, "Identification : " & frm_inicial.txt_ident.Text
    Print #1,
    Print #1, "Likelihood function"
    Print #1,
    Print #1, frm_inicial.txt_function.Text
    Print #1,
    If nac <> 0 Then
        If kc3 = 0 And kc4 = 0 Then
            Print #1, "K1 = " & frm_inicial.txt_k1.Text
            Print #1, "K2 = " & frm_inicial.txt_k2.Text
        Else:
            Print #1, "K11 = " & frm_inicial.txt_k1.Text
            Print #1, "K12 = " & frm_inicial.txt_k2.Text
            Print #1, "K21 = " & frm_inicial.txt_kc4.Text
            Print #1, "K22 = " & frm_inicial.txt_kc2.Text
        End If
    End If
    Print #1,
    Print #1, "K = " & frm_inicial.txt_kvalue.Text
    Print #1,
    Print #1, "CI95%(K) = " & frm_inicial.txt_conf1.Text & " - " &
frm_inicial.txt_conf2.Text
    Close #1
    Open CdbSaveas.FileTitle For Input As #1
    Close #1
dbErrHandler:
    Exit Sub
End Sub

Private Sub mnuHelpManual_Click()
    frm_manual.Visible = True
End Sub

Private Sub mnuHelpAbout_Click() 'acessa a tela de help
    frm_about.Visible = True
End Sub

```

FRM_SCREEN

```

Private Sub Form_Load()
    Set pic_inicial.Picture = LoadResPicture(101, vbResBitmap)
    tela = 0
End Sub

Private Sub cmd_ok2gi_Click()
    frm_screen.Visible = False: frm_2g.Visible = True: frm_2g.txt_n2g.SetFocus
    tela = 2
End Sub

Private Sub cmd_ok3gi_Click()

```

```

    frm_screen.Visible = False: frm_3g.Visible = True: frm_3g.txt_n3g.SetFocus
    tela = 3
End Sub

Private Sub cmd_ok4gi_Click()
    frm_screen.Visible = False: frm_4g.Visible = True: frm_4g.txt_n4g.SetFocus
    tela = 4
End Sub

Private Sub cmd_ok5gi_Click()
    frm_screen.Visible = False: frm_screen.Visible = False: frm_5g1.Visible = True
    tela = 5
End Sub

Private Sub cmd_okcons1_Click()
    frm_screen.Visible = False: frm_cons1.Visible = True: frm_cons1.txt_nac.SetFocus
    tela = 6
End Sub

Private Sub cmd_gemeos_Click()
    frm_screen.Visible = False: frm_gemeos.Visible = True
    tela = 7
End Sub

Private Sub cmd_back_Click()
    Call LimpaVar
    frm_inicial.Visible = True: frm_screen.Visible = False
End Sub

Private Sub cmd_enter_Click() 'retorna carregando os dados
    pen = Val(frm_screen.txt_np.Text)
    hetnpen = Val(frm_screen.txt_nhetnp.Text)
    norsf = Val(frm_screen.txt_nsf.Text)
    frm_screen.Visible = False: frm_inicial.Visible = True
    Call Run
    Call WriteFunction
    Call Confint
    frm_inicial.Label11.Visible = True: frm_inicial.Label7.Visible = True
    frm_inicial.txt_conf1.Visible = True: frm_inicial.txt_conf2.Visible = True
End Sub

```

FRM_CONS1

```

Private Sub Form_Load()
    x = 1
    txt_tree.Text = x
    Set pic_lpen.Picture = LoadResPicture(332, vbResBitmap)
    Set pic_lhetnp.Picture = LoadResPicture(333, vbResBitmap)
    Set pic_lnorsf.Picture = LoadResPicture(334, vbResBitmap)
    Set pic_2pen.Picture = LoadResPicture(332, vbResBitmap)
    Set pic_2hetnp.Picture = LoadResPicture(333, vbResBitmap)
    Set pic_2norsf.Picture = LoadResPicture(334, vbResBitmap)
    Set pic_s12g.Picture = LoadResPicture(335, vbResBitmap)
    Set pic_s13g.Picture = LoadResPicture(336, vbResBitmap)
    Set pic_s14g.Picture = LoadResPicture(337, vbResBitmap)
    Set pic_s15g.Picture = LoadResPicture(338, vbResBitmap)
    Set pic_s22g.Picture = LoadResPicture(335, vbResBitmap)
    Set pic_s23g.Picture = LoadResPicture(336, vbResBitmap)
    Set pic_s24g.Picture = LoadResPicture(337, vbResBitmap)
    Set pic_s25g.Picture = LoadResPicture(338, vbResBitmap)
    Set pic_slgem.Picture = LoadResPicture(342, vbResBitmap)
    Set pic_s2gem.Picture = LoadResPicture(342, vbResBitmap)
End Sub

Private Sub cmd_ok_Click()
    If Val(txt_nac.Text) > 2 Then
        resp = MsgBox("Maximum number of consanguineous trees is 2", 0)
        txt_nac.SetFocus
    ElseIf Val(txt_nac.Text) <> 0 Then
        cmd_ok.SetFocus
        Label12(0).Visible = True: Label13(0).Visible = True: cmd_enter.Visible = True
        pic_lpen.Visible = True: pic_lhetnp.Visible = True: pic_lnorsf.Visible = True
        pic_s12g.Visible = True: pic_s13g.Visible = True: pic_s14g.Visible = True:
        pic_s15g.Visible = True
    End If
End Sub

```



```

        txt_s12g.Visible = True: txt_s13g.Visible = True: txt_s14g.Visible = True:
txt_s15g.Visible = True
        txt_slpen.Visible = True: txt_slhetnp.Visible = True: txt_slhorsf.Visible = True:
pic_slgem.Visible = True
        Label14.Visible = True: Frame1.Visible = True: opt_sim.Visible = True:
opt_nao.Visible = True
        cmd_oks1gem.Visible = True: txt_slpen.SetFocus: txt_tree.Visible = True
    Else: frm_cons1.Visible = True
    End If
End Sub

Private Sub txt_nac_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then cmd_ok.SetFocus
End Sub

Private Sub txt_s12g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then frm_cons1.Visible = False: frm_2g.Visible = True:
frm_2g.cmd_ok2g.SetFocus
    frm_2g.txt_n2g.Text = Val(txt_s12g.Text): frm_2g.txt_n2g.Enabled = False: tela = 6
    s12g(x) = Val(txt_s12g.Text)
End Sub

Private Sub txt_s22g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then frm_cons1.Visible = False: frm_2g.Visible = True:
frm_2g.cmd_ok2g.SetFocus
    frm_2g.txt_n2g.Text = Val(txt_s22g.Text): frm_2g.txt_n2g.Enabled = False: tela = 6
    s22g(x) = Val(txt_s22g.Text)
End Sub

Private Sub txt_s13g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then frm_cons1.Visible = False: frm_3g.Visible = True:
frm_3g.cmd_ok3g.SetFocus
    frm_3g.txt_n3g.Text = Val(txt_s13g.Text): frm_3g.txt_n3g.Enabled = False: tela = 6
    s13g(x) = Val(txt_s13g.Text)
End Sub

Private Sub txt_s23g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then frm_cons1.Visible = False: frm_3g.Visible = True:
frm_3g.cmd_ok3g.SetFocus
    frm_3g.txt_n3g.Text = Val(txt_s23g.Text): frm_3g.txt_n3g.Enabled = False: tela = 6
    s23g(x) = Val(txt_s23g.Text)
End Sub

Private Sub txt_s14g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then frm_cons1.Visible = False: frm_4g.Visible = True:
frm_4g.cmd_ok4g.SetFocus
    frm_4g.txt_n4g.Text = Val(txt_s14g.Text): frm_4g.txt_n4g.Enabled = False: tela = 6
    s14g(x) = Val(txt_s14g.Text)
End Sub

Private Sub txt_s24g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then frm_cons1.Visible = False: frm_4g.Visible = True:
frm_4g.cmd_ok4g.SetFocus
    frm_4g.txt_n4g.Text = Val(txt_s24g.Text): frm_4g.txt_n4g.Enabled = False: tela = 6
    s24g(x) = Val(txt_s24g.Text)
End Sub

Private Sub txt_s15g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then frm_cons1.Visible = False: frm_5g1.Visible = True:
frm_5g1.cmd_ok5g.SetFocus
    frm_5g1.txt_n5g.Text = Val(txt_s15g.Text): frm_5g1.txt_n5g.Enabled = False: tela = 6
    s15g(x) = Val(txt_s15g.Text)
End Sub

Private Sub txt_s25g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then frm_cons1.Visible = False: frm_5g1.Visible = True:
frm_5g1.cmd_ok5g.SetFocus
    frm_5g1.txt_n5g.Text = Val(txt_s25g.Text): frm_5g1.txt_n5g.Enabled = False: tela = 6
    s25g(x) = Val(txt_s25g.Text)
End Sub

Private Sub cmd_oks1gem_Click()
    frm_cons1.Visible = False: frm_gemeos.Visible = True: tela = 6: gem = 1
    frm_gemeos.txt_gem1.SetFocus
End Sub

Private Sub cmd_oks2gem_Click()

```

```

frm_cons1.Visible = False: frm_gemeos.Visible = True: tela = 6: gem = 2
frm_gemeos.txt_gem1.SetFocus
End Sub

Private Sub opt_sim_Click()
    Label3(1).Visible = True: pic_2pen.Visible = True: pic_2hetnp.Visible = True:
    pic_2norsf.Visible = True
    pic_s22g.Visible = True: pic_s23g.Visible = True: pic_s24g.Visible = True
    txt_s2pen.Visible = True: txt_s2hetnp.Visible = True: txt_s2norsf.Visible = True
    txt_s22g.Visible = True: txt_s23g.Visible = True: txt_s24g.Visible = True
    pic_s25g.Visible = True: txt_s25g.Visible = True: pic_s2gem.Visible = True
    txt_s2pen.Text = Empty: txt_s2hetnp.Text = Empty: txt_s2norsf.Text = Empty
    txt_s22g.Text = Empty: txt_s23g.Text = Empty: txt_s24g.Text = Empty: txt_s25g.Text =
    Empty
    cmd_oks2gem.Visible = True: txt_s2pen.SetFocus: cmd_enter.Enabled = True: flag(x) =
    1
End Sub

Private Sub opt_nao_Click()
    Label3(1).Visible = True: pic_2pen.Visible = True: pic_2hetnp.Visible = True:
    pic_2norsf.Visible = True
    pic_s22g.Visible = True: pic_s23g.Visible = True: pic_s24g.Visible = True:
    pic_s25g.Visible = True
    txt_s2pen.Visible = True: txt_s2hetnp.Visible = True: txt_s2norsf.Visible = True:
    pic_s2gem.Visible = True
    txt_s22g.Visible = True: txt_s23g.Visible = True: txt_s24g.Visible = True:
    txt_s25g.Visible = True
    txt_s2pen.Text = Val(txt_slpen.Text): txt_s2hetnp.Text = Val(txt_slhetnp.Text):
    txt_s2norsf.Text = Val(txt_slhorsf.Text)
    txt_s22g.Text = Val(txt_sl2g.Text): txt_s23g.Text = Val(txt_sl3g.Text)
    txt_s24g.Text = Val(txt_sl4g.Text): txt_s25g.Text = Val(txt_sl5g.Text)
    cmd_oks2gem.Visible = True: cmd_enter.Enabled = True: cmd_enter.SetFocus: flag(x) =
    0
End Sub

Private Sub txt_s2norsf_KeyPress(KeyAscii As Integer)
    If KeyAscii = 8 Then cmd_enter.SetFocus
End Sub

Private Sub cmd_back_Click()
    txt_slpen.Text = Empty: txt_slhetnp.Text = Empty: txt_slhorsf.Text = Empty
    txt_s2pen.Text = Empty: txt_s2hetnp.Text = Empty: txt_s2norsf.Text = Empty
    txt_sl2g.Text = Empty: txt_sl3g.Text = Empty: txt_sl4g.Text = Empty: txt_sl5g.Text =
    Empty
    txt_s22g.Text = Empty: txt_s23g.Text = Empty: txt_s24g.Text = Empty: txt_s25g.Text =
    Empty
    txt_nac.Text = Empty: txt_tree.Text = Empty
    frm_2g.txt_n2g.Enabled = True: frm_3g.txt_n3g.Enabled = True: frm_4g.txt_n4g.Enabled
    = True: frm_5g.txt_n5g.Enabled = True
    Label2(0).Visible = False: Label3(0).Visible = False: cmd_enter.Visible = False
    pic_lpen.Visible = False: pic_lhetnp.Visible = False: pic_lhorsf.Visible = False
    pic_sl2g.Visible = False: pic_sl3g.Visible = False: pic_sl4g.Visible = False:
    pic_sl5g.Visible = False
    txt_sl2g.Visible = False: txt_sl3g.Visible = False: txt_sl4g.Visible = False:
    txt_sl5g.Visible = False
    txt_slpen.Visible = False: txt_slhetnp.Visible = False: txt_slhorsf.Visible = False:
    pic_slgem.Visible = False
    Label4.Visible = False: Frame1.Visible = False: opt_sim.Visible = False:
    opt_nao.Visible = False
    cmd_oks1gem.Visible = False: txt_tree.Visible = False
    opt_sim.Value = False: opt_nao.Value = False: cmd_enter.Enabled = False
    txt_s2pen.Visible = False: txt_s2hetnp.Visible = False: txt_s2norsf.Visible = False
    txt_s22g.Visible = False: txt_s23g.Visible = False: txt_s24g.Visible = False:
    txt_s25g.Visible = False
    pic_2pen.Visible = False: pic_2hetnp.Visible = False: pic_2norsf.Visible = False:
    Label3(1).Visible = False
    pic_s22g.Visible = False: pic_s23g.Visible = False: pic_s24g.Visible = False:
    pic_s25g.Visible = False
    pic_s2gem.Visible = False: cmd_oks2gem.Visible = False
    tela = 0: gem = 0: frm_cons1.Visible = False: frm_screen.Visible = True
End Sub

Private Sub cmd_enter_Click()
    nac = Val(txt_nac.Text)
    slpen(x) = Val(txt_slpen.Text): slhet(x) = Val(txt_slhetnp.Text): slhorsf(x) =
    Val(txt_slhorsf.Text)

```

```

s2pen(x) = Val(txt_s2pen.Text): s2het(x) = Val(txt_s2hetnp.Text): s2nsf(x) =
Val(txt_s2norsf.Text)
s12g(x) = Val(txt_s12g.Text): s13g(x) = Val(txt_s13g.Text): s14g(x) =
Val(txt_s14g.Text): s15g(x) = Val(txt_s15g.Text)
s22g(x) = Val(txt_s22g.Text): s23g(x) = Val(txt_s23g.Text): s24g(x) =
Val(txt_s24g.Text): s25g(x) = Val(txt_s25g.Text)
If flag(x) = 0 Then
    For i = 1 To s12g(x): s22gi(x, i) = s12gi(x, i): Next i
    For i = 1 To s13g(x)
        s23gi(x, i) = s13gi(x, i)
        For j = 1 To s13gi(x, i): s23gij(x, i, j) = s13gij(x, i, j): Next j
    Next i
    For i = 1 To s14g(x)
        s24gi(x, i) = s14gi(x, i)
        For j = 1 To s14gi(x, i)
            s24gij(x, i, j) = s14gij(x, i, j)
            For l = 1 To s14gij(x, i, j): s24gijl(x, i, j, l) = s14gijl(x, i, j, l): Next
1
        Next j
    Next i
    For i = 1 To s15g(x)
        s25gi(x, i) = s15gi(x, i)
        For j = 1 To s15gi(x, i)
            s25gij(x, i, j) = s15gij(x, i, j)
            For l = 1 To s15gij(x, i, j)
                s25gijl(x, i, j, l) = s15gijl(x, i, j, l)
                For m = 1 To s15gijl(x, i, j, l): s25gijlm(x, i, j, l, m) = s15gijlm(x, i,
j, l, m): Next m
            Next l
        Next j
    Next i
    s2gem1(x) = s1gem1(x): s2gem2(x) = s1gem2(x): s2gem3(x) = s1gem3(x)
End If
'limpa as caixas de texto
txt_nac.Enabled = False: x = x + 1: txt_tree.Text = x
txt_slpen.Text = Empty: txt_slhetnp.Text = Empty: txt_slhorsf.Text = Empty
txt_s2pen.Text = Empty: txt_s2hetnp.Text = Empty: txt_s2norsf.Text = Empty
txt_s12g.Text = Empty: txt_s13g.Text = Empty: txt_s14g.Text = Empty: txt_s15g.Text =
Empty
txt_s22g.Text = Empty: txt_s23g.Text = Empty: txt_s24g.Text = Empty: txt_s25g.Text =
Empty
opt_sim.Value = False: opt_ao.Value = False: txt_slpen.SetFocus: cmd_enter.Enabled
= False
txt_s2pen.Visible = False: txt_s2hetnp.Visible = False: txt_s2norsf.Visible = False
txt_s22g.Visible = False: txt_s23g.Visible = False: txt_s24g.Visible = False:
txt_s25g.Visible = False
pic_2pen.Visible = False: pic_2hetnp.Visible = False: pic_2norsf.Visible = False:
Label3(1).Visible = False
pic_s22g.Visible = False: pic_s23g.Visible = False: pic_s24g.Visible = False:
pic_s25g.Visible = False
pic_s2gem.Visible = False: cmd_oks2gem.Visible = False
If x > Val(txt_nac.Text) Then
    txt_nac.Text = Empty: txt_nac.Enabled = True: txt_tree.Text = Empty
    frm_2g.txt_n2g.Enabled = True: frm_3g.txt_n3g.Enabled = True
    frm_4g.txt_n4g.Enabled = True: frm_5g1.txt_n5g.Enabled = True
    Label2(0).Visible = False: Label3(0).Visible = False: cmd_enter.Visible = False
    pic_lpen.Visible = False: pic_lhetnp.Visible = False: pic_lhorsf.Visible = False
    pic_sl2g.Visible = False: pic_sl3g.Visible = False: pic_sl4g.Visible = False:
pic_sl5g.Visible = False
    txt_sl2g.Visible = False: txt_sl3g.Visible = False: txt_sl4g.Visible = False:
txt_sl5g.Visible = False
    txt_slpen.Visible = False: txt_slhetnp.Visible = False: txt_slhorsf.Visible =
False: pic_slgem.Visible = False
    Label4.Visible = False: Frame1.Visible = False: opt_sim.Visible = False:
opt_ao.Visible = False
    cmd_okslgem.Visible = False: txt_tree.Visible = False
    frm_screen.Visible = True: frm_consl.Visible = False: tela = 0: gem = 0
End If
End Sub

```

FRM_GEMEOS

```

Private Sub Form_Load()
    Set pic_gem1.Picture = LoadResPicture(342, vbResBitmap)

```

```

    Set pic_gem2.Picture = LoadResPicture(340, vbResBitmap)
    Set pic_gem3.Picture = LoadResPicture(341, vbResBitmap)
End Sub

Private Sub cmd_back_Click()
    txt_gem1.Text = Empty: txt_gem2.Text = Empty: txt_gem3.Text = Empty
    If tela = 7 Then tela = 0: gem = 0: frm_gemeos.Visible = False:
    frm_screen.Visible = True
    If tela = 6 Then tela = 0: gem = 0: frm_gemeos.Visible = False:
    frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    If tela = 7 Then
        ngem1 = Val(txt_gem1.Text): ngem2 = Val(txt_gem2.Text): ngem3 =
        Val(txt_gem3.Text)
        frm_gemeos.Visible = False: frm_screen.Visible = True
    ElseIf tela = 6 Then
        If gem = 2 Then
            s2gem1(x) = Val(txt_gem1.Text): s2gem2(x) = Val(txt_gem2.Text):
            s2gem3(x) = Val(txt_gem3.Text)
            Else: slgem1(x) = Val(txt_gem1.Text): slgem2(x) = Val(txt_gem2.Text):
            slgem3(x) = Val(txt_gem3.Text)
            End If
        frm_gemeos.Visible = False: frm_cons1.Visible = True
    End If
    txt_gem1.Text = Empty: txt_gem2.Text = Empty: txt_gem3.Text = Empty
End Sub

```

FRM_WRITE

```

Private Sub cmd_back_Click()
    frm_write.Visible = False
End Sub

Private Sub cmd_save_Click()
    cdbSave.Flags = &H400 + &H2: cdbSave.Filter = "Text docs (*.txt)|*.txt"
    cdbSave.DefaultExt = "txt": cdbSave.ShowSave
    On Error GoTo dbErrHandler
    Open cdbSave.FileName For Output As #1
    Print #1, frm_write.txt_write.Text
    Close #1
    Open cdbSave.FileName For Input As #1
    Call ReadFile
    Call Run
    Call WriteFunction
    Call Confint
    frm_inicial.Label1.Visible = True: frm_inicial.Label7.Visible = True
    frm_inicial.txt_conf1.Visible = True: frm_inicial.txt_conf2.Visible = True
dbErrHandler:
End Sub

```

FRM_MANUAL

```

Private Sub cmd_ok_Click()
    frm_manual.Visible = False
End Sub

```

FRM_ABOUT

```

Const READ_CONTROL = &H20000
Const KEY_QUERY_VALUE = &H1
Const KEY_SET_VALUE = &H2
Const KEY_CREATE_SUB_KEY = &H4
Const KEY_ENUMERATE_SUB_KEYS = &H8
Const KEY_NOTIFY = &H10
Const KEY_CREATE_LINK = &H20

```

```

Const KEY_ALL_ACCESS = KEY_QUERY_VALUE + KEY_SET_VALUE + _
                        KEY_CREATE_SUB_KEY + KEY_ENUMERATE_SUB_KEYS + _
                        KEY_NOTIFY + KEY_CREATE_LINK + READ_CONTROL
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1
Const REG_DWORD = 4
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"

Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey
As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long,
ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA"
(ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef
lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long

Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub

Private Sub cmdOK_Click()
    Unload Me
End Sub

Public Sub StartSysInfo()
    On Error GoTo SysInfoErr
    Dim rc As Long
    Dim SysInfoPath As String
    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO, SysInfoPath) Then
    ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC,
SysInfoPath) Then
        If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
            SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
        Else
            GoTo SysInfoErr
        End If
    Else
        GoTo SysInfoErr
    End If

    Call Shell(SysInfoPath, vbNormalFocus)
    Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As String,
ByRef KeyVal As String) As Boolean
    Dim i As Long
    Dim rc As Long
    Dim hKey As Long
    Dim hDepth As Long
    Dim KeyValType As Long
    Dim tmpVal As String
    Dim KeyValSize As Long
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey)
    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError
    tmpVal = String$(1024, 0)
    KeyValSize = 1024
    rc = RegQueryValueEx(hKey, SubKeyRef, 0, _
                        KeyValType, tmpVal, KeyValSize)
    If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError
    If (Asc(Mid(tmpVal, KeyValSize, 1)) = 0) Then
        tmpVal = Left(tmpVal, KeyValSize - 1)
    Else
        tmpVal = Left(tmpVal, KeyValSize)
    End If
    Select Case KeyValType
    Case REG_SZ
        KeyVal = tmpVal
    Case REG_DWORD
        For i = Len(tmpVal) To 1 Step -1
            KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))
        Next i
    End Select
    GetKeyValue = True
GetKeyError:
    GetKeyValue = False
End Function

```

```

        Next i
        KeyVal = Format$("&h" + KeyVal)
    End Select
    GetKeyValue = True
    rc = RegCloseKey(hKey)
    Exit Function
GetKeyError:
    KeyVal = ""
    GetKeyValue = False
    rc = RegCloseKey(hKey)
End Function

```

FRM_2G

```

Private Sub txt_n2g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Or KeyAscii = 8 Then cmd_ok2g.SetFocus
End Sub

Private Sub cmd_ok2g_Click()
    If Val(txt_n2g.Text) > 15 Then
        resp = MsgBox("Maximum number of two-generations normal trees is 15", 0)
        txt_n2g.SetFocus
    ElseIf Val(txt_n2g.Text) <> 0 Then
        cmd_ok2g.SetFocus
        For i = 1 To Val(txt_n2g.Text)
            Label2.Visible = True: Label3(i - 1).Visible = True
            txt_2g(i - 1).Visible = True: txt_2g(0).SetFocus
        Next i
        cmd_enter.Visible = True
    Else: frm_2g.Visible = True
    End If
End Sub

Private Sub txt_2g_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_n2g.Text)
            txt_2g(i - 1).Visible = True
            If i = Val(txt_n2g.Text) Then cmd_enter.SetFocus
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_n2g.Text = Empty
    For i = 1 To 15
        Label3(i - 1).Visible = False: txt_2g(i - 1).Text = Empty: txt_2g(i - 1).Visible =
False
    Next i
    Label2.Visible = False: cmd_enter.Visible = False
    If tela = 2 Then tela = 0: frm_2g.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_2g.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", vbYesNo)
    If (resp = 6) Then
        If tela = 2 Then
            n2g = Val(txt_n2g.Text)
            For i = 1 To 15: n2gi(i) = Val(txt_2g(i - 1).Text): Next i
            frm_screen.Visible = True: frm_2g.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s22g.Text) <> 0 Then
                For i = 1 To s22g(x): s22gi(x, i) = Val(txt_2g(i - 1).Text): Next i
            Else: For i = 1 To s12g(x): s12gi(x, i) = Val(txt_2g(i - 1).Text): Next i
            End If
            frm_cons1.Visible = True: frm_2g.Visible = False
        End If
        txt_n2g.Text = Empty: tela = 0
        For i = 1 To 15
            Label3(i - 1).Visible = False: txt_2g(i - 1).Text = Empty: txt_2g(i - 1).Visible
= False
        Next i
        Label2.Visible = False: cmd_enter.Visible = False
    ElseIf (resp = 7) Then frm_2g.Visible = True
    End If

```

End Sub

FRM_3G

```
Private Sub Form_Load()  
    Picture1.Move 120, 240, 7815, 2.75 * ScaleHeight  
    txt_n3g.Move 2400, 0, 375, 315  
    cont = 840  
    ctr = 360  
    For i = 1 To 10  
        txt_3gn2(i - 1).Move 4440, 480 + (i - 1) * cont, 375, 315  
    Next i  
    For i = 1 To 10: txt_3gn3(i - 1).Move 3480 + (i - 1) * ctr, 840, 375, 315: Next i  
    For i = 11 To 20: txt_3gn3(i - 1).Move 3480 + (i - 11) * ctr, 1680, 375, 315: Next i  
    For i = 21 To 30: txt_3gn3(i - 1).Move 3480 + (i - 21) * ctr, 2520, 375, 315: Next i  
    For i = 31 To 40: txt_3gn3(i - 1).Move 3480 + (i - 31) * ctr, 3360, 375, 315: Next i  
    For i = 41 To 50: txt_3gn3(i - 1).Move 3480 + (i - 41) * ctr, 4200, 375, 315: Next i  
    For i = 51 To 60: txt_3gn3(i - 1).Move 3480 + (i - 51) * ctr, 5040, 375, 315: Next i  
    For i = 61 To 70: txt_3gn3(i - 1).Move 3480 + (i - 61) * ctr, 5880, 375, 315: Next i  
    For i = 71 To 80: txt_3gn3(i - 1).Move 3480 + (i - 71) * ctr, 6720, 375, 315: Next i  
    For i = 81 To 90: txt_3gn3(i - 1).Move 3480 + (i - 81) * ctr, 7560, 375, 315: Next i  
    For i = 91 To 100: txt_3gn3(i - 1).Move 3480 + (i - 91) * ctr, 8400, 375, 315: Next  
i  
End Sub  
  
Sub Form_Resize()  
    HScroll11.Left = 0  
    VScroll13g.Top = 0  
    If Picture1.Width > ScaleWidth Then  
        HScroll11.Top = ScaleHeight - HScroll11.Height  
    Else  
        HScroll11.Top = ScaleHeight  
    End If  
    If Picture1.Height > HScroll11.Top Then  
        VScroll13g.Left = ScaleWidth - VScroll13g.Width  
        If Picture1.Width > VScroll13g.Left Then  
            HScroll11.Top = ScaleHeight - HScroll11.Height  
        End If  
    Else  
        VScroll13g.Left = ScaleWidth  
    End If  
    HScroll11.Width = ScaleWidth  
    If HScroll11.Top > 0 Then VScroll13g.Height = HScroll11.Top  
    HScroll11.Max = Picture1.Width - VScroll13g.Left  
    VScroll13g.Max = Picture1.Height - HScroll11.Top  
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1  
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1  
    VScroll13g.SmallChange = Abs(VScroll13g.Max \ 16) + 1  
    VScroll13g.LargeChange = Abs(VScroll13g.Max \ 4) + 1  
    HScroll11.ZOrder 0  
    VScroll13g.ZOrder 0  
End Sub  
  
Sub HScroll13g_Change()  
    Picture1.Left = -HScroll11.Value  
End Sub  
  
Sub VScroll13g_Change()  
    Picture1.Top = -VScroll13g.Value  
End Sub  
  
Private Sub txt_n3g_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Or KeyAscii = 8 Then cmd_ok3g.SetFocus  
End Sub  
  
Private Sub cmd_ok3g_Click()  
    If Val(txt_n3g.Text) > 10 Then  
        resp = MsgBox("Maximum number of three-generations normal trees is 10", 0)  
        txt_n3g.SetFocus  
    ElseIf Val(txt_n3g.Text) <= 10 And Val(txt_n3g.Text) <> 0 Then  
        cmd_ok3g.SetFocus  
        If Val(txt_n3g.Text) > 3 Then VScroll13g.Visible = True  
        If Val(txt_n3g.Text) <> 0 Then  
            For i = 1 To Val(txt_n3g.Text)
```

```

        label2(i - 1).Visible = True: txt_3gn2(i - 1).Visible = True:
txt_3gn2(0).SetFocus
    Next i
End If
Else: frm_3g.Visible = True
End If
End Sub

Private Sub txt_3gn2_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_n3g.Text)
            If Val(txt_3gn2(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the second-generation is 10",
0)
                txt_3gn2(i - 1).SetFocus
            ElseIf Val(txt_3gn2(i - 1).Text) <= 10 And Val(txt_3gn2(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_3gn2(i - 1).Text)
                    Label3(i - 1).Visible = True
                    txt_3gn3(j - 1 + 10 * (i - 1)).Visible = True
                    If j = 1 Then txt_3gn3(j - 1 + 10 * (i - 1)).SetFocus
                    If i = Val(txt_n3g.Text) And j = Val(txt_3gn2(i - 1).Text) Then
cmd_enter.Enabled = True
                    Next j
                End If
            End If
        Next i
    End If
End Sub

Private Sub txt_3gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_n3g.Text)
            For j = 1 To Val(txt_3gn2(i - 1).Text)
                txt_3gn3(j - 1 + 10 * (i - 1)).Visible = True
                If i = n3g And j = Val(txt_3gn2(i - 1).Text) Then cmd_back.SetFocus
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_n3g.Text = Empty: cmd_enter.Enabled = False
    For i = 1 To 10
        txt_3gn2(i - 1).Text = Empty: label2(i - 1).Visible = False
        Label3(i - 1).Visible = False: txt_3gn2(i - 1).Visible = False
    Next i
    For i = 1 To 100: txt_3gn3(i - 1).Text = Empty: txt_3gn3(i - 1).Visible = False:
Next i
    If tela = 3 Then tela = 0: frm_3g.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_3g.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 3 Then
            n3g = Val(txt_n3g.Text)
            For i = 1 To 10
                n3gi(i) = Val(txt_3gn2(i - 1).Text)
                For j = 1 To 10: n3gij(i, j) = Val(txt_3gn3(j - 1 + 10 * (i - 1)).Text): Next
j
            Next i
            frm_screen.Visible = True: frm_3g.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s23g.Text) <> 0 Then
                For i = 1 To s23g(x)
                    s23gi(x, i) = Val(txt_3gn2(i - 1).Text)
                    For j = 1 To s23gi(x, i): s23gij(x, i, j) = Val(txt_3gn3(j - 1 + 10 * (i -
1)).Text): Next j
                Next i
            Else:
                For i = 1 To s13g(x)
                    s13gi(x, i) = Val(txt_3gn2(i - 1).Text)
                    For j = 1 To s13gi(x, i): s13gij(x, i, j) = Val(txt_3gn3(j - 1 + 10 * (i -
1)).Text): Next j
                Next i
            End If
        End If
    End Sub

```



```

        frm_cons1.Visible = True: frm_3g.Visible = False
    End If
    txt_n3g.Text = Empty: tela = 0: cmd_enter.Enabled = False
    For i = 1 To 10
        txt_3gn2(i - 1).Text = Empty: label2(i - 1).Visible = False
        Label3(i - 1).Visible = False: txt_3gn2(i - 1).Visible = False
    Next i
    For i = 1 To 100: txt_3gn3(i - 1).Text = Empty: txt_3gn3(i - 1).Visible = False:
Next i
    ElseIf (resp = 7) Then frm_3g.Visible = True
    End If
End Sub

```

FRM_4G

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8535, 2.5 * ScaleHeight
    txt_n4g.Move 2280, 120, 375, 285
    txt_4gn2.Move 4440, 480, 375, 285
    cont = 720
    ctr = 360
    For i = 1 To 10
        txt_4gn3(i - 1).Move 3240, 840 + (i - 1) * cont, 375, 285
    Next i
    For i = 1 To 10: txt_4gn4(i - 1).Move 4680 + (i - 1) * ctr, 1200, 375, 285: Next i
    For i = 11 To 20: txt_4gn4(i - 1).Move 4680 + (i - 11) * ctr, 1920, 375, 285: Next i
    For i = 21 To 30: txt_4gn4(i - 1).Move 4680 + (i - 21) * ctr, 2640, 375, 285: Next i
    For i = 31 To 40: txt_4gn4(i - 1).Move 4680 + (i - 31) * ctr, 3360, 375, 285: Next i
    For i = 41 To 50: txt_4gn4(i - 1).Move 4680 + (i - 41) * ctr, 4080, 375, 285: Next i
    For i = 51 To 60: txt_4gn4(i - 1).Move 4680 + (i - 51) * ctr, 4800, 375, 285: Next i
    For i = 61 To 70: txt_4gn4(i - 1).Move 4680 + (i - 61) * ctr, 5520, 375, 285: Next i
    For i = 71 To 80: txt_4gn4(i - 1).Move 4680 + (i - 71) * ctr, 6240, 375, 285: Next i
    For i = 81 To 90: txt_4gn4(i - 1).Move 4680 + (i - 81) * ctr, 6960, 375, 285: Next i
    For i = 91 To 100: txt_4gn4(i - 1).Move 4680 + (i - 91) * ctr, 7680, 375, 285: Next
i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll14g.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll14g.Left = ScaleWidth - VScroll14g.Width
        If Picture1.Width > VScroll14g.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll14g.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll14g.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll14g.Left
    VScroll14g.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll14g.SmallChange = Abs(VScroll14g.Max \ 16) + 1
    VScroll14g.LargeChange = Abs(VScroll14g.Max \ 4) + 1
    HScroll11.ZOrder 0
    VScroll14g.ZOrder 0
End Sub

Sub HScroll14g_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g_Change()
    Picture1.Top = -VScroll14g.Value
End Sub

Private Sub txt_n4g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Or KeyAscii = 8 Then cmd_ok4g.SetFocus

```

```

End Sub

Private Sub cmd_ok4g_Click()
    If Val(txt_n4g.Text) > 10 Then
        resp = MsgBox("Maximum number of four-generations normal trees is 10", 0)
        txt_n4g.SetFocus
    ElseIf Val(txt_n4g.Text) <= 10 And Val(txt_n4g.Text) <> 0 Then
        txt_4gn2.Visible = True: txt_4gn2.SetFocus
        label2(0).Visible = True
    Else: frm_4g.Visible = True
    End If
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_4gn2.Text) > 10 Then
            resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
            txt_4gn2.SetFocus
        ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then
            For i = 1 To Val(txt_4gn2.Text)
                txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_4gn3(i - 1).SetFocus
            Next i
            If Val(txt_4gn2.Text) > 3 Then VScroll4g.Visible = True
        End If
    End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_4gn2.Text)
            If Val(txt_4gn3(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
                txt_4gn3(i - 1).SetFocus
            ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_4gn3(i - 1).Text)
                    txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
                    If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
                Next j
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.Enabled = True
                End If
            End If
        Next i
    End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click() 'retorna sem guardar dados
    label2(0).Visible = False: txt_4gn2.Text = Empty: txt_4gn2.Visible = False
    cmd_enter.Enabled = False: txt_n4g.Text = Empty
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click() 'retorna carregando os valores informados
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then

```

```

n4g = Val(txt_n4g.Text)
n4gi(1) = Val(txt_4gn2.Text)
For j = 1 To 10
    n4gij(1, j) = Val(txt_4gn3(j - 1).Text)
    For l = 1 To 10: n4gijl(1, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
    Next j
    frm_screen.Visible = True: frm_4g.Visible = False
ElseIf tela = 6 Then
    If Val(frm_cons1.txt_s24g.Text) <> 0 Then
        s24gi(x, 1) = Val(txt_4gn2.Text)
        For j = 1 To s24gi(x, 1)
            s24gij(x, 1, j) = Val(txt_4gn3(j - 1).Text)
            For l = 1 To s24gij(x, 1, j): s24gijl(x, 1, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
        Next j
    Else:
        s14gi(x, 1) = Val(txt_4gn2.Text)
        For j = 1 To s14gi(x, 1)
            s14gij(x, 1, j) = Val(txt_4gn3(j - 1).Text)
            For l = 1 To s14gij(x, 1, j): s14gijl(x, 1, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
        Next j
        frm_cons1.Visible = True: frm_4g.Visible = False
    End If
End If
'limpa as caixas de texto
label2(0).Visible = False: txt_4gn2.Text = Empty: txt_4gn2.Visible = False:
cmd_enter.Enabled = False
For i = 1 To 10
    Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
    txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
Next i
For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If Val(txt_n4g.Text) > 1 Then
        frm_4gl.Visible = True: frm_4gl.txt_4gn2.SetFocus: frm_4g.Visible = False
    Else
        If tela = 4 Then tela = 0: txt_n4g.Text = Empty: frm_4g.Visible = False:
        frm_screen.Visible = True
        If tela = 6 Then tela = 0: txt_n4g.Text = Empty: frm_4g.Visible = False:
        frm_cons1.Visible = True
    End If
    ElseIf (resp = 7) Then frm_4g.Visible = True
    End If
End Sub

```

FRM_4G1

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight
    txt_4gn2.Move 4440, 120, 375, 285
    cont = 720
    ctr = 360
    For i = 1 To 10
        txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285
    Next i
    For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i
    For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i
    For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i
    For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i
    For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i
    For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i
    For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i
    For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i
    For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i
    For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next
i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll14gl.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    End If
End Sub

```

```

Else
    HScroll11.Top = ScaleHeight
End If
If Picture1.Height > HScroll11.Top Then
    VScroll14g1.Left = ScaleWidth - VScroll14g1.Width
    If Picture1.Width > VScroll14g1.Left Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    End If
Else
    VScroll14g1.Left = ScaleWidth
End If
HScroll11.Width = ScaleWidth
If HScroll11.Top > 0 Then VScroll14g1.Height = HScroll11.Top
HScroll11.Max = Picture1.Width - VScroll14g1.Left
VScroll14g1.Max = Picture1.Height - HScroll11.Top
HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
VScroll14g1.SmallChange = Abs(VScroll14g1.Max \ 16) + 1
VScroll14g1.LargeChange = Abs(VScroll14g1.Max \ 4) + 1
HScroll11.ZOrder 0
VScroll14g1.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g1_Change()
    Picture1.Top = -VScroll14g1.Value
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_4gn2.Text) > 10 Then
            resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
            txt_4gn2.SetFocus
        ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then
            For i = 1 To Val(txt_4gn2.Text)
                txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_4gn3(i - 1).SetFocus
            Next i
            If Val(txt_4gn2.Text) > 3 Then VScroll14g1.Visible = True
        End If
    End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_4gn2.Text)
            If Val(txt_4gn3(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
                txt_4gn3(i - 1).SetFocus
            ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_4gn3(i - 1).Text)
                    txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
                    If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
                    If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                        cmd_enter.Enabled = True
                    End If
                Next j
            End If
        Next i
    End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

```

```

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g1.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g1.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(2) = Val(txt_4gn2.Text)
            For j = 1 To 10
                n4gij(2, j) = Val(txt_4gn3(j - 1).Text)
                For l = 1 To 10: n4gijl(2, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
            Next j
            frm_screen.Visible = True: frm_4g1.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s24g.Text) <> 0 Then
                s24gi(x, 2) = Val(txt_4gn2.Text)
                For j = 1 To s24gi(x, 2)
                    s24gij(x, 2, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s24gij(x, 2, j): s24gijl(x, 2, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                Next j
            Else:
                s14gi(x, 2) = Val(txt_4gn2.Text)
                For j = 1 To s14gi(x, 2)
                    s14gij(x, 2, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s14gij(x, 2, j): s14gijl(x, 2, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                Next j
                frm_cons1.Visible = True: frm_4g1.Visible = False
            End If
        End If
        txt_4gn2.Text = Empty
        For i = 1 To 10
            Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
            txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
        Next i
        For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
        If Val(frm_4g.txt_n4g.Text) > 2 Then
            frm_4g2.Visible = True: frm_4g2.txt_4gn2.SetFocus: frm_4g1.Visible = False
        Else
            If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g1.Visible = False:
frm_screen.Visible = True
            If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g1.Visible = False:
frm_cons1.Visible = True
        End If
        ElseIf (resp = 7) Then frm_4g1.Visible = True
        End If
    End Sub

```

FRM_4G2

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight
    txt_4gn2.Move 4440, 120, 375, 285
    cont = 720
    ctr = 360
    For i = 1 To 10
        txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285
    Next i
    For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i
    For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i
    For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i
    For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i

```

```

For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i
For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i
For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i
For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i
For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i
For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next
i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll14g2.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll14g2.Left = ScaleWidth - VScroll14g2.Width
        If Picture1.Width > VScroll14g2.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll14g2.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll14g2.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll14g2.Left
    VScroll14g2.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll14g2.SmallChange = Abs(VScroll14g2.Max \ 16) + 1
    VScroll14g2.LargeChange = Abs(VScroll14g2.Max \ 4) + 1
    HScroll11.ZOrder 0
    VScroll14g2.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g2_Change()
    Picture1.Top = -VScroll14g2.Value
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_4gn2.Text) > 10 Then
            resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
            txt_4gn2.SetFocus
        ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then
            For i = 1 To Val(txt_4gn2.Text)
                txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_4gn3(i - 1).SetFocus
            Next i
            If Val(txt_4gn2.Text) > 3 Then VScroll14g2.Visible = True
        End If
    End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_4gn2.Text)
            If Val(txt_4gn3(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
                txt_4gn3(i - 1).SetFocus
            ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_4gn3(i - 1).Text)
                    txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
                    If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
                    If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
cmd_enter.Enabled = True
                    Next j
                End If
            Next i
        End If
    End If
End Sub

```

```

End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g2.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g2.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(3) = Val(txt_4gn2.Text)
            For j = 1 To 10
                n4gij(3, j) = Val(txt_4gn3(j - 1).Text)
                For l = 1 To 10: n4gijl(3, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
            Next j
            frm_screen.Visible = True: frm_4g2.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s24g.Text) <> 0 Then
                s24gi(x, 3) = Val(txt_4gn2.Text)
                For j = 1 To s24gi(x, 3)
                    s24gij(x, 3, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s24gij(x, 3, j): s24gijl(x, 3, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                Next j
            Else:
                s14gi(x, 3) = Val(txt_4gn2.Text)
                For j = 1 To s14gi(x, 3)
                    s14gij(x, 3, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s14gij(x, 3, j): s14gijl(x, 3, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                Next j
                frm_cons1.Visible = True: frm_4g2.Visible = False
            End If
        End If
        txt_4gn2.Text = Empty
        For i = 1 To 10
            Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
            txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
        Next i
        For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
        If Val(frm_4g.txt_n4g.Text) > 3 Then
            frm_4g3.Visible = True: frm_4g3.txt_4gn2.SetFocus: frm_4g2.Visible = False
        Else
            If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g2.Visible = False:
            frm_screen.Visible = True
            If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g2.Visible = False:
            frm_cons1.Visible = True
        End If
        ElseIf (resp = 7) Then frm_4g2.Visible = True
    End If
End Sub

```

FRM_4G3

```
Private Sub Form_Load()  
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight  
    txt_4gn2.Move 4440, 120, 375, 285  
    cont = 720  
    ctr = 360  
    For i = 1 To 10  
        txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285  
    Next i  
    For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i  
    For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i  
    For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i  
    For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i  
    For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i  
    For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i  
    For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i  
    For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i  
    For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i  
    For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next  
i  
End Sub  
  
Sub Form_Resize()  
    HScroll11.Left = 0  
    VScroll4g3.Top = 0  
    If Picture1.Width > ScaleWidth Then  
        HScroll11.Top = ScaleHeight - HScroll11.Height  
    Else  
        HScroll11.Top = ScaleHeight  
    End If  
    If Picture1.Height > HScroll11.Top Then  
        VScroll4g3.Left = ScaleWidth - VScroll4g3.Width  
        If Picture1.Width > VScroll4g3.Left Then  
            HScroll11.Top = ScaleHeight - HScroll11.Height  
        End If  
    Else  
        VScroll4g3.Left = ScaleWidth  
    End If  
    HScroll11.Width = ScaleWidth  
    If HScroll11.Top > 0 Then VScroll4g3.Height = HScroll11.Top  
    HScroll11.Max = Picture1.Width - VScroll4g3.Left  
    VScroll4g3.Max = Picture1.Height - HScroll11.Top  
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1  
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1  
    VScroll4g3.SmallChange = Abs(VScroll4g3.Max \ 16) + 1  
    VScroll4g3.LargeChange = Abs(VScroll4g3.Max \ 4) + 1  
    HScroll11.ZOrder 0  
    VScroll4g3.ZOrder 0  
End Sub  
  
Sub HScroll11_Change()  
    Picture1.Left = -HScroll11.Value  
End Sub  
  
Sub VScroll4g3_Change()  
    Picture1.Top = -VScroll4g3.Value  
End Sub  
  
Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)  
    If KeyAscii = 13 Then  
        If Val(txt_4gn2.Text) > 10 Then  
            resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)  
            txt_4gn2.SetFocus  
        ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then  
            For i = 1 To Val(txt_4gn2.Text)  
                txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True  
                If i = 1 Then txt_4gn3(i - 1).SetFocus  
            Next i  
            If Val(txt_4gn2.Text) > 3 Then VScroll4g3.Visible = True  
        End If  
    End If  
End Sub  
  
Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)  
    If KeyAscii = 13 Then
```



```

For i = 1 To Val(txt_4gn2.Text)
    If Val(txt_4gn3(i - 1).Text) > 10 Then
        resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
        txt_4gn3(i - 1).SetFocus
    ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
        For j = 1 To Val(txt_4gn3(i - 1).Text)
            txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
            If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
            If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
cmd_enter.Enabled = True
                Next j
            End If
        Next i
    End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g3.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g3.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(4) = Val(txt_4gn2.Text)
            For j = 1 To 10
                n4gij(4, j) = Val(txt_4gn3(j - 1).Text)
                For l = 1 To 10: n4gijl(4, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
            Next j
            frm_screen.Visible = True: frm_4g3.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s24g.Text) <> 0 Then
                s24gi(x, 4) = Val(txt_4gn2.Text)
                For j = 1 To s24gi(x, 4)
                    s24gij(x, 4, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s24gij(x, 4, j): s24gijl(x, 4, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                Next j
            Else:
                s14gi(x, 4) = Val(txt_4gn2.Text)
                For j = 1 To s14gi(x, 4)
                    s14gij(x, 4, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s14gij(x, 4, j): s14gijl(x, 4, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                Next j
                frm_cons1.Visible = True: frm_4g3.Visible = False
            End If
        End If
        txt_4gn2.Text = Empty
        For i = 1 To 10
            Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
            txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
        Next i
    End If
End Sub

```

```

    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If Val(frm_4g.txt_n4g.Text) > 4 Then
        frm_4g4.Visible = True: frm_4g4.txt_4gn2.SetFocus: frm_4g3.Visible = False
    Else
        If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g3.Visible = False:
        frm_screen.Visible = True
        If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g3.Visible = False:
        frm_cons1.Visible = True
    End If
    ElseIf (resp = 7) Then frm_4g3.Visible = True
    End If
End Sub

```

FRM_4G4

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight
    txt_4gn2.Move 4440, 120, 375, 285
    cont = 720
    ctr = 360
    For i = 1 To 10
        txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285
    Next i
    For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i
    For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i
    For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i
    For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i
    For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i
    For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i
    For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i
    For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i
    For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i
    For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next
i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll14g4.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll14g4.Left = ScaleWidth - VScroll14g4.Width
        If Picture1.Width > VScroll14g4.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll14g4.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll14g4.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll14g4.Left
    VScroll14g4.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll14g4.SmallChange = Abs(VScroll14g4.Max \ 16) + 1
    VScroll14g4.LargeChange = Abs(VScroll14g4.Max \ 4) + 1
    HScroll11.ZOrder 0
    VScroll14g4.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g4_Change()
    Picture1.Top = -VScroll14g4.Value
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then

```

```

    If Val(txt_4gn2.Text) > 10 Then
        resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
        txt_4gn2.SetFocus
    ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then
        For i = 1 To Val(txt_4gn2.Text)
            txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
            If i = 1 Then txt_4gn3(i - 1).SetFocus
        Next i
        If Val(txt_4gn2.Text) > 3 Then VScroll4g4.Visible = True
    End If
End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_4gn2.Text)
            If Val(txt_4gn3(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
                txt_4gn3(i - 1).SetFocus
            ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_4gn3(i - 1).Text)
                    txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
                    If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
                    If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                        cmd_enter.Enabled = True
                    End If
                Next j
            End If
        Next i
    End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g4.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g4.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(5) = Val(txt_4gn2.Text)
            For j = 1 To 10
                n4gij(5, j) = Val(txt_4gn3(j - 1).Text)
                For l = 1 To 10: n4gijl(5, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
                Next j
                frm_screen.Visible = True: frm_4g4.Visible = False
            ElseIf tela = 6 Then
                If Val(frm_cons1.txt_s24g.Text) <> 0 Then
                    s24gi(x, 5) = Val(txt_4gn2.Text)
                    For j = 1 To s24gi(x, 5)
                        s24gij(x, 5, j) = Val(txt_4gn3(j - 1).Text)
                        For l = 1 To s24gij(x, 5, j): s24gijl(x, 5, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                    Next j
                Else:

```

```

        s14gi(x, 5) = Val(txt_4gn2.Text)
        For j = 1 To s14gi(x, 5)
            s14gij(x, 5, j) = Val(txt_4gn3(j - 1).Text)
            For l = 1 To s14gij(x, 5, j): s14gijl(x, 5, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
        Next j
        frm_cons1.Visible = True: frm_4g4.Visible = False
    End If
End If
txt_4gn2.Text = Empty
For i = 1 To 10
    Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
    txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
Next i
For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
If Val(frm_4g.txt_n4g.Text) > 5 Then
    frm_4g5.Visible = True: frm_4g5.txt_4gn2.SetFocus: frm_4g4.Visible = False
Else
    If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g4.Visible = False:
frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g4.Visible = False:
frm_cons1.Visible = True
    End If
    ElseIf (resp = 7) Then frm_4g4.Visible = True
    End If
End Sub

```

FRM_4G5

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight
    txt_4gn2.Move 4440, 120, 375, 285
    cont = 720
    ctr = 360
    For i = 1 To 10
        txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285
    Next i
    For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i
    For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i
    For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i
    For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i
    For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i
    For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i
    For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i
    For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i
    For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i
    For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next
i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll14g5.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll14g5.Left = ScaleWidth - VScroll14g5.Width
        If Picture1.Width > VScroll14g5.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll14g5.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll14g5.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll14g5.Left
    VScroll14g5.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll14g5.SmallChange = Abs(VScroll14g5.Max \ 16) + 1
    VScroll14g5.LargeChange = Abs(VScroll14g5.Max \ 4) + 1

```

```

HScroll11.ZOrder 0
VScroll14g5.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g5_Change()
    Picture1.Top = -VScroll14g5.Value
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_4gn2.Text) > 10 Then
            resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
            txt_4gn2.SetFocus
        ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then
            For i = 1 To Val(txt_4gn2.Text)
                txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_4gn3(i - 1).SetFocus
            Next i
            If Val(txt_4gn2.Text) > 3 Then VScroll14g5.Visible = True
        End If
    End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_4gn2.Text)
            If Val(txt_4gn3(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
                txt_4gn3(i - 1).SetFocus
            ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_4gn3(i - 1).Text)
                    txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
                    If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
                    If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                        cmd_enter.Enabled = True
                    End If
                Next j
            End If
        Next i
    End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g5.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g5.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(6) = Val(txt_4gn2.Text)
            For j = 1 To 10

```

```

        n4gij(6, j) = Val(txt_4gn3(j - 1).Text)
        For l = 1 To 10: n4gijl(6, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next j
    Next j
    frm_screen.Visible = True: frm_4g5.Visible = False
ElseIf tela = 6 Then
    If Val(frm_cons1.txt_s24g.Text) <> 0 Then
        s24gi(x, 6) = Val(txt_4gn2.Text)
        For j = 1 To s24gi(x, 6)
            s24gij(x, 6, j) = Val(txt_4gn3(j - 1).Text)
            For l = 1 To s24gij(x, 6, j): s24gijl(x, 6, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
        Next j
    Else:
        s14gi(x, 6) = Val(txt_4gn2.Text)
        For j = 1 To s14gi(x, 6)
            s14gij(x, 6, j) = Val(txt_4gn3(j - 1).Text)
            For l = 1 To s14gij(x, 6, j): s14gijl(x, 6, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
        Next j
        frm_cons1.Visible = True: frm_4g5.Visible = False
    End If
End If
txt_4gn2.Text = Empty
For i = 1 To 10
    Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
    txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
Next i
For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If Val(frm_4g.txt_n4g.Text) > 6 Then
        frm_4g6.Visible = True: frm_4g6.txt_4gn2.SetFocus: frm_4g5.Visible = False
    Else
        If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g5.Visible = False:
        frm_screen.Visible = True
        If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g5.Visible = False:
        frm_cons1.Visible = True
    End If
    ElseIf (resp = 7) Then frm_4g5.Visible = True
End If
End Sub

```

FRM_4G6

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight
    txt_4gn2.Move 4440, 120, 375, 285
    cont = 720
    ctr = 360
    For i = 1 To 10
        txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285
    Next i
    For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i
    For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i
    For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i
    For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i
    For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i
    For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i
    For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i
    For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i
    For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i
    For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next
i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll14g6.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll14g6.Left = ScaleWidth - VScroll14g6.Width
    End If
End Sub

```

```

    If Picture1.Width > VScroll14g6.Left Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    End If
Else
    VScroll14g6.Left = ScaleWidth
End If
HScroll11.Width = ScaleWidth
If HScroll11.Top > 0 Then VScroll14g6.Height = HScroll11.Top
HScroll11.Max = Picture1.Width - VScroll14g6.Left
VScroll14g6.Max = Picture1.Height - HScroll11.Top
HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
VScroll14g6.SmallChange = Abs(VScroll14g6.Max \ 16) + 1
VScroll14g6.LargeChange = Abs(VScroll14g6.Max \ 4) + 1
HScroll11.ZOrder 0
VScroll14g6.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g6_Change()
    Picture1.Top = -VScroll14g6.Value
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_4gn2.Text) > 10 Then
            resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
            txt_4gn2.SetFocus
        ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then
            For i = 1 To Val(txt_4gn2.Text)
                txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_4gn3(i - 1).SetFocus
            Next i
            If Val(txt_4gn2.Text) > 3 Then VScroll14g6.Visible = True
        End If
    End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_4gn2.Text)
            If Val(txt_4gn3(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
                txt_4gn3(i - 1).SetFocus
            ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_4gn3(i - 1).Text)
                    txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
                    If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
                    If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                        cmd_enter.Enabled = True
                    End If
                Next j
            End If
        Next i
    End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
    Next j
End Sub

```

```

        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g6.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g6.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(7) = Val(txt_4gn2.Text)
            For j = 1 To 10
                n4gij(7, j) = Val(txt_4gn3(j - 1).Text)
                For l = 1 To 10: n4gijl(7, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
                Next j
                frm_screen.Visible = True: frm_4g6.Visible = False
            ElseIf tela = 6 Then
                If Val(frm_cons1.txt_s24g.Text) <> 0 Then
                    s24gi(x, 7) = Val(txt_4gn2.Text)
                    For j = 1 To s24gi(x, 7)
                        s24gij(x, 7, j) = Val(txt_4gn3(j - 1).Text)
                        For l = 1 To s24gij(x, 7, j): s24gijl(x, 7, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                    Next j
                Else:
                    s14gi(x, 7) = Val(txt_4gn2.Text)
                    For j = 1 To s14gi(x, 7)
                        s14gij(x, 7, j) = Val(txt_4gn3(j - 1).Text)
                        For l = 1 To s14gij(x, 7, j): s14gijl(x, 7, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                    Next j
                    frm_cons1.Visible = True: frm_4g6.Visible = False
                End If
            End If
            'limpa as caixas de texto
            txt_4gn2.Text = Empty
            For i = 1 To 10
                Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
                txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
            Next i
            For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
            If Val(frm_4g.txt_n4g.Text) > 7 Then
                frm_4g7.Visible = True: frm_4g7.txt_4gn2.SetFocus: frm_4g6.Visible = False
            Else
                If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g6.Visible = False:
frm_screen.Visible = True
                If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g6.Visible = False:
frm_cons1.Visible = True
            End If
            ElseIf (resp = 7) Then frm_4g6.Visible = True
            End If
        End Sub

```

FRM_4G7

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight
    txt_4gn2.Move 4440, 120, 375, 285
    cont = 720
    ctr = 360
    For i = 1 To 10
        txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285
    Next i
    For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i
    For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i
    For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i
    For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i
    For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i
    For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i
    For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i
    For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i

```



```

For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i
For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next
i
End Sub

Sub Form_Resize()
HScroll11.Left = 0
VScroll14g7.Top = 0
If Picture1.Width > ScaleWidth Then
HScroll11.Top = ScaleHeight - HScroll11.Height
Else
HScroll11.Top = ScaleHeight
End If
If Picture1.Height > HScroll11.Top Then
VScroll14g7.Left = ScaleWidth - VScroll14g7.Width
If Picture1.Width > VScroll14g7.Left Then
HScroll11.Top = ScaleHeight - HScroll11.Height
End If
Else
VScroll14g7.Left = ScaleWidth
End If
HScroll11.Width = ScaleWidth
If HScroll11.Top > 0 Then VScroll14g7.Height = HScroll11.Top
HScroll11.Max = Picture1.Width - VScroll14g7.Left
VScroll14g7.Max = Picture1.Height - HScroll11.Top
HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
VScroll14g7.SmallChange = Abs(VScroll14g7.Max \ 16) + 1
VScroll14g7.LargeChange = Abs(VScroll14g7.Max \ 4) + 1
HScroll11.ZOrder 0
VScroll14g7.ZOrder 0
End Sub

Sub HScroll11_Change()
Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g7_Change()
Picture1.Top = -VScroll14g7.Value
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then
If Val(txt_4gn2.Text) > 10 Then
resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
txt_4gn2.SetFocus
ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then
For i = 1 To Val(txt_4gn2.Text)
txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
If i = 1 Then txt_4gn3(i - 1).SetFocus
Next i
If Val(txt_4gn2.Text) > 3 Then VScroll14g7.Visible = True
End If
End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
If KeyAscii = 13 Then
For i = 1 To Val(txt_4gn2.Text)
If Val(txt_4gn3(i - 1).Text) > 10 Then
resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
txt_4gn3(i - 1).SetFocus
ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
For j = 1 To Val(txt_4gn3(i - 1).Text)
txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
cmd_enter.Enabled = True
Next j
End If
Next i
End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
If KeyAscii = 9 Then

```

```

For i = 1 To Val(txt_4gn2.Text)
    For j = 1 To Val(txt_4gn3(i - 1).Text)
        txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
        If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
            cmd_enter.SetFocus
        End If
    Next j
Next i
End If
End Sub

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g7.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g7.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(8) = Val(txt_4gn2.Text)
            For j = 1 To 10
                n4gij(8, j) = Val(txt_4gn3(j - 1).Text)
                For l = 1 To 10: n4gijl(8, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
                Next j
                frm_screen.Visible = True: frm_4g7.Visible = False
            ElseIf tela = 6 Then
                If Val(frm_cons1.txt_s24g.Text) <> 0 Then
                    s24gi(x, 8) = Val(txt_4gn2.Text)
                    For j = 1 To s24gi(x, 8)
                        s24gij(x, 8, j) = Val(txt_4gn3(j - 1).Text)
                        For l = 1 To s24gij(x, 8, j): s24gijl(x, 8, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                    Next j
                Else:
                    s14gi(x, 8) = Val(txt_4gn2.Text)
                    For j = 1 To s14gi(x, 8)
                        s14gij(x, 8, j) = Val(txt_4gn3(j - 1).Text)
                        For l = 1 To s14gij(x, 8, j): s14gijl(x, 8, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                    Next j
                    frm_cons1.Visible = True: frm_4g7.Visible = False
                End If
            End If
            txt_4gn2.Text = Empty
            For i = 1 To 10
                Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
                txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
            Next i
            For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
            If Val(frm_4g.txt_n4g.Text) > 8 Then
                frm_4g8.Visible = True: frm_4g8.txt_4gn2.SetFocus: frm_4g7.Visible = False
            Else
                If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g7.Visible = False:
frm_screen.Visible = True
                If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g7.Visible = False:
frm_cons1.Visible = True
            End If
            ElseIf (resp = 7) Then frm_4g7.Visible = True
        End If
    End Sub

```

FRM_4G8

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight

```

```

txt_4gn2.Move 4440, 120, 375, 285
cont = 720
ctr = 360
For i = 1 To 10
    txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285
Next i
For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i
For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i
For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i
For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i
For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i
For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i
For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i
For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i
For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i
For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next
i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll14g8.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll14g8.Left = ScaleWidth - VScroll14g8.Width
        If Picture1.Width > VScroll14g8.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll14g8.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll14g8.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll14g8.Left
    VScroll14g8.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll14g8.SmallChange = Abs(VScroll14g8.Max \ 16) + 1
    VScroll14g8.LargeChange = Abs(VScroll14g8.Max \ 4) + 1
    HScroll11.ZOrder 0
    VScroll14g8.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g8_Change()
    Picture1.Top = -VScroll14g8.Value
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_4gn2.Text) > 10 Then
            resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
            txt_4gn2.SetFocus
        ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then
            For i = 1 To Val(txt_4gn2.Text)
                txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_4gn3(i - 1).SetFocus
            Next i
            If Val(txt_4gn2.Text) > 3 Then VScroll14g8.Visible = True
        End If
    End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_4gn2.Text)
            If Val(txt_4gn3(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)

```

```

        txt_4gn3(i - 1).SetFocus
    ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
        For j = 1 To Val(txt_4gn3(i - 1).Text)
            txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
            If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
            If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                cmd_enter.Enabled = True
            End If
        Next j
    End If
Next i
End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g8.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g8.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(9) = Val(txt_4gn2.Text)
            For j = 1 To 10
                n4gij(9, j) = Val(txt_4gn3(j - 1).Text)
                For l = 1 To 10: n4gijl(9, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
            Next j
            frm_screen.Visible = True: frm_4g8.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s24g.Text) <> 0 Then
                s24gi(x, 9) = Val(txt_4gn2.Text)
                For j = 1 To s24gi(x, 9)
                    s24gij(x, 9, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s24gij(x, 9, j): s24gijl(x, 9, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                Next j
            Else:
                s14gi(x, 9) = Val(txt_4gn2.Text)
                For j = 1 To s14gi(x, 9)
                    s14gij(x, 9, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s14gij(x, 9, j): s14gijl(x, 9, j, l) = Val(txt_4gn4(l - 1 + 10
* (j - 1)).Text): Next l
                Next j
                frm_cons1.Visible = True: frm_4g8.Visible = False
            End If
        End If
        'limpa as caixas de texto
        txt_4gn2.Text = Empty
        For i = 1 To 10
            Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
            txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
        Next i
        For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
        If Val(frm_4g.txt_n4g.Text) > 9 Then
            frm_4g9.Visible = True: frm_4g9.txt_4gn2.SetFocus: frm_4g8.Visible = False

```

```

Else
    If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g8.Visible = False:
    frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g8.Visible = False:
    frm_cons1.Visible = True
End If
ElseIf (resp = 7) Then frm_4g8.Visible = True
End If
End Sub

```

FRM_4G9

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 8415, 2.8 * ScaleHeight
    txt_4gn2.Move 4440, 120, 375, 285
    cont = 720
    ctr = 360
    For i = 1 To 10
        txt_4gn3(i - 1).Move 2760, 480 + (i - 1) * cont, 375, 285
    Next i
    For i = 1 To 10: txt_4gn4(i - 1).Move 4320 + (i - 1) * ctr, 840, 375, 285: Next i
    For i = 11 To 20: txt_4gn4(i - 1).Move 4320 + (i - 11) * ctr, 1560, 375, 285: Next i
    For i = 21 To 30: txt_4gn4(i - 1).Move 4320 + (i - 21) * ctr, 2280, 375, 285: Next i
    For i = 31 To 40: txt_4gn4(i - 1).Move 4320 + (i - 31) * ctr, 3000, 375, 285: Next i
    For i = 41 To 50: txt_4gn4(i - 1).Move 4320 + (i - 41) * ctr, 3720, 375, 285: Next i
    For i = 51 To 60: txt_4gn4(i - 1).Move 4320 + (i - 51) * ctr, 4440, 375, 285: Next i
    For i = 61 To 70: txt_4gn4(i - 1).Move 4320 + (i - 61) * ctr, 5160, 375, 285: Next i
    For i = 71 To 80: txt_4gn4(i - 1).Move 4320 + (i - 71) * ctr, 5880, 375, 285: Next i
    For i = 81 To 90: txt_4gn4(i - 1).Move 4320 + (i - 81) * ctr, 6600, 375, 285: Next i
    For i = 91 To 100: txt_4gn4(i - 1).Move 4320 + (i - 91) * ctr, 7320, 375, 285: Next
i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll14g9.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll14g9.Left = ScaleWidth - VScroll14g9.Width
        If Picture1.Width > VScroll14g9.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll14g9.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll14g9.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll14g9.Left
    VScroll14g9.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll14g9.SmallChange = Abs(VScroll14g9.Max \ 16) + 1
    VScroll14g9.LargeChange = Abs(VScroll14g9.Max \ 4) + 1
    HScroll11.ZOrder 0
    VScroll14g9.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll14g9_Change()
    Picture1.Top = -VScroll14g9.Value
End Sub

Private Sub txt_4gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_4gn2.Text) > 10 Then
            resp = MsgBox("Maximum number of individuals in the second-generation is 10", 0)
            txt_4gn2.SetFocus
        ElseIf Val(txt_4gn2.Text) <= 10 And Val(txt_4gn2.Text) <> 0 Then

```

```

        For i = 1 To Val(txt_4gn2.Text)
            txt_4gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
            If i = 1 Then txt_4gn3(i - 1).SetFocus
        Next i
        If Val(txt_4gn2.Text) > 3 Then VScroll4g9.Visible = True
    End If
End If
End Sub

Private Sub txt_4gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_4gn2.Text)
            If Val(txt_4gn3(i - 1).Text) > 10 Then
                resp = MsgBox("Maximum number of individuals in the third-generation is 10",
0)
                txt_4gn3(i - 1).SetFocus
            ElseIf Val(txt_4gn3(i - 1).Text) <= 10 And Val(txt_4gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_4gn3(i - 1).Text)
                    txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True: Label4(i - 1).Visible = True
                    If j = 1 Then txt_4gn4(j - 1 + 10 * (i - 1)).SetFocus
                    If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                        cmd_enter.Enabled = True
                    End If
                Next j
            End If
        Next i
    End If
End Sub

Private Sub txt_4gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_4gn2.Text)
            For j = 1 To Val(txt_4gn3(i - 1).Text)
                txt_4gn4(j - 1 + 10 * (i - 1)).Visible = True
                If i = Val(txt_4gn2.Text) And j = Val(txt_4gn3(i - 1).Text) Then
                    cmd_enter.SetFocus
                End If
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()
    txt_4gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 10
        Label3(j - 1).Visible = False: Label4(j - 1).Visible = False
        txt_4gn3(j - 1).Text = Empty: txt_4gn3(j - 1).Visible = False
    Next j
    For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
    If tela = 4 Then tela = 0: frm_4g9.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_4g9.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 4 Then
            n4gi(10) = Val(txt_4gn2.Text)
            For j = 1 To 10
                n4gij(10, j) = Val(txt_4gn3(j - 1).Text)
                For l = 1 To 10: n4gijl(10, j, l) = Val(txt_4gn4(l - 1 + 10 * (j - 1)).Text):
Next l
            Next j
            frm_screen.Visible = True: frm_4g9.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s24g.Text) <> 0 Then
                s24gi(x, 10) = Val(txt_4gn2.Text)
                For j = 1 To s24gi(x, 10)
                    s24gij(x, 10, j) = Val(txt_4gn3(j - 1).Text)
                    For l = 1 To s24gij(x, 10, j): s24gijl(x, 10, j, l) = Val(txt_4gn4(l - 1 +
10 * (j - 1)).Text): Next l
                Next j
            Else:
                s14gi(x, 10) = Val(txt_4gn2.Text)
                For j = 1 To s14gi(x, 10)
                    s14gij(x, 10, j) = Val(txt_4gn3(j - 1).Text)

```

```

        For l = 1 To s14gij(x, 10, j): s14gijl(x, 10, j, l) = Val(txt_4gn4(l - 1 +
10 * (j - 1)).Text): Next l
    Next j
    frm_cons1.Visible = True: frm_4g9.Visible = False
End If
End If
txt_4gn2.Text = Empty
For i = 1 To 10
    Label3(i - 1).Visible = False: Label4(i - 1).Visible = False
    txt_4gn3(i - 1).Text = Empty: txt_4gn3(i - 1).Visible = False
Next i
For i = 1 To 100: txt_4gn4(i - 1).Text = Empty: txt_4gn4(i - 1).Visible = False:
Next i
If tela = 4 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g9.Visible = False:
frm_screen.Visible = True
If tela = 6 Then tela = 0: frm_4g.txt_n4g.Text = Empty: frm_4g9.Visible = False:
frm_cons1.Visible = True
ElseIf (resp = 7) Then frm_4g9.Visible = True
End If
End Sub

```

FRM_5G1

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 2.2 * ScaleWidth, 2.5 * ScaleHeight
    txt_n5g.Move 2280, 0, 375, 285
    txt_5gn2.Move 4680, 480, 375, 285
    txt_5gn3(0).Move 4680, 840, 375, 285: txt_5gn3(1).Move 4680, 3720, 375, 285
    txt_5gn3(2).Move 4680, 6600, 375, 285: txt_5gn3(3).Move 12120, 840, 375, 285
    txt_5gn3(4).Move 12120, 3720, 375, 285: txt_5gn3(5).Move 12120, 6600, 375, 285
    ctr = 360
    For i = 1 To 6
        txt_5gn4(i - 1).Move 3120, 1200 + (i - 1) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 1) * ctr, 1200, 375, 285
    Next i
    For i = 7 To 12
        txt_5gn4(i - 1).Move 3120, 4080 + (i - 7) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 7) * ctr, 1560, 375, 285
    Next i
    For i = 13 To 18
        txt_5gn4(i - 1).Move 3120, 6960 + (i - 13) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 13) * ctr, 1920, 375, 285
    Next i
    For i = 19 To 24
        txt_5gn4(i - 1).Move 10560, 1200 + (i - 19) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 19) * ctr, 2280, 375, 285
    Next i
    For i = 25 To 30
        txt_5gn4(i - 1).Move 10560, 4080 + (i - 25) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 25) * ctr, 2640, 375, 285
    Next i
    For i = 31 To 36
        txt_5gn4(i - 1).Move 10560, 6960 + (i - 31) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 31) * ctr, 3000, 375, 285
    Next i
    For i = 37 To 42: txt_5gn5(i - 1).Move 5040 + (i - 37) * ctr, 4080, 375, 285: Next i
    For i = 43 To 48: txt_5gn5(i - 1).Move 5040 + (i - 43) * ctr, 4440, 375, 285: Next i
    For i = 49 To 54: txt_5gn5(i - 1).Move 5040 + (i - 49) * ctr, 4800, 375, 285: Next i
    For i = 55 To 60: txt_5gn5(i - 1).Move 5040 + (i - 55) * ctr, 5160, 375, 285: Next i
    For i = 61 To 66: txt_5gn5(i - 1).Move 5040 + (i - 61) * ctr, 5520, 375, 285: Next i
    For i = 67 To 72: txt_5gn5(i - 1).Move 5040 + (i - 67) * ctr, 5880, 375, 285: Next i
    For i = 73 To 78: txt_5gn5(i - 1).Move 5040 + (i - 73) * ctr, 6960, 375, 285: Next i
    For i = 79 To 84: txt_5gn5(i - 1).Move 5040 + (i - 79) * ctr, 7320, 375, 285: Next i
    For i = 85 To 90: txt_5gn5(i - 1).Move 5040 + (i - 85) * ctr, 7680, 375, 285: Next i
    For i = 91 To 96: txt_5gn5(i - 1).Move 5040 + (i - 91) * ctr, 8040, 375, 285: Next i
    For i = 97 To 102: txt_5gn5(i - 1).Move 5040 + (i - 97) * ctr, 8400, 375, 285: Next
i
    For i = 103 To 108: txt_5gn5(i - 1).Move 5040 + (i - 103) * ctr, 8760, 375, 285:
Next i
    For i = 109 To 114: txt_5gn5(i - 1).Move 12480 + (i - 109) * ctr, 1200, 375,
285:Next i
    For i = 115 To 120: txt_5gn5(i - 1).Move 12480 + (i - 115) * ctr, 1560, 375,
285:Next i
    For i = 121 To 126: txt_5gn5(i - 1).Move 12480 + (i - 121) * ctr, 1920, 375,
285:Next i

```

```

    For i = 127 To 132: txt_5gn5(i - 1).Move 12480 + (i - 127) * ctr, 2280, 375,
285:Next i
    For i = 133 To 138: txt_5gn5(i - 1).Move 12480 + (i - 133) * ctr, 2640, 375,
285:Next i
    For i = 139 To 144: txt_5gn5(i - 1).Move 12480 + (i - 139) * ctr, 3000, 375,
285:Next i
    For i = 145 To 150: txt_5gn5(i - 1).Move 12480 + (i - 145) * ctr, 4080, 375,
285:Next i
    For i = 151 To 156: txt_5gn5(i - 1).Move 12480 + (i - 151) * ctr, 4440, 375,
285:Next i
    For i = 157 To 162: txt_5gn5(i - 1).Move 12480 + (i - 157) * ctr, 4800, 375,
285:Next i
    For i = 163 To 168: txt_5gn5(i - 1).Move 12480 + (i - 163) * ctr, 5160, 375,
285:Next i
    For i = 169 To 174: txt_5gn5(i - 1).Move 12480 + (i - 169) * ctr, 5520, 375,
285:Next i
    For i = 175 To 180: txt_5gn5(i - 1).Move 12480 + (i - 175) * ctr, 5880, 375,
285:Next i
    For i = 181 To 186: txt_5gn5(i - 1).Move 12480 + (i - 181) * ctr, 6960, 375,
285:Next i
    For i = 187 To 192: txt_5gn5(i - 1).Move 12480 + (i - 187) * ctr, 7320, 375,
285:Next i
    For i = 193 To 198: txt_5gn5(i - 1).Move 12480 + (i - 193) * ctr, 7680, 375,
285:Next i
    For i = 199 To 204: txt_5gn5(i - 1).Move 12480 + (i - 199) * ctr, 8040, 375,
285:Next i
    For i = 205 To 210: txt_5gn5(i - 1).Move 12480 + (i - 205) * ctr, 8400, 375,
285:Next i
    For i = 211 To 216: txt_5gn5(i - 1).Move 12480 + (i - 211) * ctr, 8760, 375,
285:Next i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll11.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll11.Left = ScaleWidth - VScroll11.Width
        If Picture1.Width > VScroll11.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll11.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll11.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll11.Left
    VScroll11.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll11.SmallChange = Abs(VScroll11.Max \ 16) + 1
    VScroll11.LargeChange = Abs(VScroll11.Max \ 4) + 1
    HScroll11.ZOrder 0
    VScroll11.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll11_Change()
    Picture1.Top = -VScroll11.Value
End Sub

Private Sub txt_n5g_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Or KeyAscii = 8 Then cmd_ok5g.SetFocus
End Sub

Private Sub cmd_ok5g_Click()
    If Val(txt_n5g.Text) > 6 Then
        resp = MsgBox("Maximum number of five-generations trees is 6", 0)
        txt_n5g.SetFocus
    ElseIf Val(txt_n5g.Text) <= 6 And Val(txt_n5g.Text) <> 0 Then

```



```

        txt_5gn2.Visible = True: txt_5gn2.SetFocus: label2(0).Visible = True
    Else: frm_5g1.Visible = True
    End If
End Sub

Private Sub txt_5gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_5gn2.Text) > 6 Then
            resp = MsgBox("Maximum number of individuals in the second generation is 6", 0)
            txt_5gn2.SetFocus
        ElseIf Val(txt_5gn2.Text) <= 6 And Val(txt_5gn2.Text) <> 0 Then
            For i = 1 To Val(txt_5gn2.Text)
                txt_5gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_5gn3(i - 1).SetFocus
            Next i
            If Val(txt_5gn2.Text) > 1 And Val(txt_5gn2.Text) <= 3 Then VScroll11.Visible = True
            If Val(txt_5gn2.Text) > 3 Then HScroll11.Visible = True: VScroll11.Visible = True
        End If
    End If
End Sub

Private Sub txt_5gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            If Val(txt_5gn3(i - 1).Text) > 6 Then
                resp = MsgBox("Maximum number of individuals in the third generation is 6", 0)
                txt_5gn3(i - 1).SetFocus
            ElseIf Val(txt_5gn3(i - 1).Text) <= 6 And Val(txt_5gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_5gn3(i - 1).Text)
                    txt_5gn4(j - 1 + 6 * (i - 1)).Visible = True: Label5(j - 1 + 6 * (i - 1)).Visible = True
                    If j = 1 Then txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                Next j
            End If
        Next i
    End If
End Sub

Private Sub txt_5gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            For j = 1 To Val(txt_5gn3(i - 1).Text)
                If Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) > 6 Then
                    resp = MsgBox("Maximum number of individuals in the fourth generation is 6", 0)
                    txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                ElseIf Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) <= 6 And Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) <> 0 Then
                    For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                        txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True: Label4(j - 1 + 6 * (i - 1)).Visible = True
                        If l = 1 Then txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).SetFocus
                    Next l
                    If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l = Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.Enabled = True
                End If
            Next j
        Next i
    End If
End Sub

Private Sub txt_5gn5_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 9 Then
        For i = 1 To Val(txt_5gn2.Text)
            For j = 1 To Val(txt_5gn3(i - 1).Text)
                For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                    txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True
                Next l
                If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l = Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.SetFocus
            Next j
        Next i
    End If
End Sub

Private Sub cmd_back_Click()

```

```

label2(0).Visible = False: txt_5gn2.Text = Empty: txt_5gn2.Visible = False
cmd_enter.Enabled = False: txt_n5g.Text = Empty
For j = 1 To 6: Label3(j - 1).Visible = False: txt_5gn3(j - 1).Text = Empty:
txt_5gn3(j - 1).Visible = False: Next j
For i = 1 To 36
    Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
    txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
Next i
For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
If tela = 5 Then tela = 0: frm_5gl.Visible = False: frm_screen.Visible = True
If tela = 6 Then tela = 0: frm_5gl.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 5 Then
            n5g = Val(frm_5gl.txt_n5g.Text)
            n5gi(1) = Val(txt_5gn2.Text)
            For j = 1 To n5gi(1)
                n5gij(1, j) = Val(txt_5gn3(j - 1).Text)
                For l = 1 To n5gij(1, j)
                    n5gijl(1, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                    For m = 1 To n5gijl(1, j, l)
                        n5gijlm(1, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                    Next m
                Next l
            Next j
            frm_screen.Visible = True: frm_5gl.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s25g.Text) <> 0 Then
                s25gi(x, 1) = Val(txt_5gn2.Text)
                For j = 1 To s25gi(x, 1)
                    s25gij(x, 1, j) = Val(txt_5gn3(j - 1).Text)
                    For l = 1 To s25gij(x, 1, j)
                        s25gijl(x, 1, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                        For m = 1 To s25gijl(x, 1, j, l)
                            s25gijlm(x, 1, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                        Next m
                    Next l
                Next j
            Else:
                s15gi(x, 1) = Val(txt_5gn2.Text)
                For j = 1 To s15gi(x, 1)
                    s15gij(x, 1, j) = Val(txt_5gn3(j - 1).Text)
                    For l = 1 To s15gij(x, 1, j)
                        s15gijl(x, 1, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                        For m = 1 To s15gijl(x, 1, j, l)
                            s15gijlm(x, 1, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                        Next m
                    Next l
                Next j
                frm_cons1.Visible = True: frm_5gl.Visible = False
            End If
        End If
        txt_5gn2.Text = Empty: cmd_enter.Enabled = False
        For j = 1 To 6
            Label3(j - 1).Visible = False: txt_5gn3(j - 1).Text = Empty: txt_5gn3(j -
1).Visible = False
        Next j
        For i = 1 To 36
            Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
            txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
        Next i
        For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
        Next i
        If Val(frm_5gl.txt_n5g.Text) > 1 Then
            frm_5gl.Visible = False: frm_5g2.Visible = True: frm_5g2.txt_5gn2.SetFocus
        Else
            If tela = 5 Then tela = 0: frm_5gl.txt_n5g.Text = Empty: frm_5gl.Visible =
False: frm_screen.Visible = True
            If tela = 6 Then tela = 0: frm_5gl.txt_n5g.Text = Empty: frm_5gl.Visible =
False: frm_cons1.Visible = True
        End If
    End If

```

```

End If
ElseIf (resp = 7) Then frm_5g1.Visible = True
End If
End Sub

```

FRM_5G2

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 2.2 * ScaleWidth, 2.5 * ScaleHeight
    txt_5gn2.Move 4680, 480, 375, 285
    txt_5gn3(0).Move 4680, 840, 375, 285: txt_5gn3(1).Move 4680, 3720, 375, 285
    txt_5gn3(2).Move 4680, 6600, 375, 285: txt_5gn3(3).Move 12120, 840, 375, 285
    txt_5gn3(4).Move 12120, 3720, 375, 285: txt_5gn3(5).Move 12120, 6600, 375, 285
    ctr = 360
    For i = 1 To 6
        txt_5gn4(i - 1).Move 3120, 1200 + (i - 1) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 1) * ctr, 1200, 375, 285
    Next i
    For i = 7 To 12
        txt_5gn4(i - 1).Move 3120, 4080 + (i - 7) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 7) * ctr, 1560, 375, 285
    Next i
    For i = 13 To 18
        txt_5gn4(i - 1).Move 3120, 6960 + (i - 13) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 13) * ctr, 1920, 375, 285
    Next i
    For i = 19 To 24
        txt_5gn4(i - 1).Move 10560, 1200 + (i - 19) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 19) * ctr, 2280, 375, 285
    Next i
    For i = 25 To 30
        txt_5gn4(i - 1).Move 10560, 4080 + (i - 25) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 25) * ctr, 2640, 375, 285
    Next i
    For i = 31 To 36
        txt_5gn4(i - 1).Move 10560, 6960 + (i - 31) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 31) * ctr, 3000, 375, 285
    Next i
    For i = 37 To 42: txt_5gn5(i - 1).Move 5040 + (i - 37) * ctr, 4080, 375, 285: Next i
    For i = 43 To 48: txt_5gn5(i - 1).Move 5040 + (i - 43) * ctr, 4440, 375, 285: Next i
    For i = 49 To 54: txt_5gn5(i - 1).Move 5040 + (i - 49) * ctr, 4800, 375, 285: Next i
    For i = 55 To 60: txt_5gn5(i - 1).Move 5040 + (i - 55) * ctr, 5160, 375, 285: Next i
    For i = 61 To 66: txt_5gn5(i - 1).Move 5040 + (i - 61) * ctr, 5520, 375, 285: Next i
    For i = 67 To 72: txt_5gn5(i - 1).Move 5040 + (i - 67) * ctr, 5880, 375, 285: Next i
    For i = 73 To 78: txt_5gn5(i - 1).Move 5040 + (i - 73) * ctr, 6960, 375, 285: Next i
    For i = 79 To 84: txt_5gn5(i - 1).Move 5040 + (i - 79) * ctr, 7320, 375, 285: Next i
    For i = 85 To 90: txt_5gn5(i - 1).Move 5040 + (i - 85) * ctr, 7680, 375, 285: Next i
    For i = 91 To 96: txt_5gn5(i - 1).Move 5040 + (i - 91) * ctr, 8040, 375, 285: Next i
    For i = 97 To 102: txt_5gn5(i - 1).Move 5040 + (i - 97) * ctr, 8400, 375, 285: Next i
    For i = 103 To 108: txt_5gn5(i - 1).Move 5040 + (i - 103) * ctr, 8760, 375, 285:
    Next i
    For i = 109 To 114: txt_5gn5(i - 1).Move 12480 + (i - 109) * ctr, 1200, 375,
285:Next i
    For i = 115 To 120: txt_5gn5(i - 1).Move 12480 + (i - 115) * ctr, 1560, 375,
285:Next i
    For i = 121 To 126: txt_5gn5(i - 1).Move 12480 + (i - 121) * ctr, 1920, 375,
285:Next i
    For i = 127 To 132: txt_5gn5(i - 1).Move 12480 + (i - 127) * ctr, 2280, 375,
285:Next i
    For i = 133 To 138: txt_5gn5(i - 1).Move 12480 + (i - 133) * ctr, 2640, 375,
285:Next i
    For i = 139 To 144: txt_5gn5(i - 1).Move 12480 + (i - 139) * ctr, 3000, 375,
285:Next i
    For i = 145 To 150: txt_5gn5(i - 1).Move 12480 + (i - 145) * ctr, 4080, 375,
285:Next i
    For i = 151 To 156: txt_5gn5(i - 1).Move 12480 + (i - 151) * ctr, 4440, 375,
285:Next i
    For i = 157 To 162: txt_5gn5(i - 1).Move 12480 + (i - 157) * ctr, 4800, 375,
285:Next i
    For i = 163 To 168: txt_5gn5(i - 1).Move 12480 + (i - 163) * ctr, 5160, 375,
285:Next i
    For i = 169 To 174: txt_5gn5(i - 1).Move 12480 + (i - 169) * ctr, 5520, 375,
285:Next i

```

```

    For i = 175 To 180: txt_5gn5(i - 1).Move 12480 + (i - 175) * ctr, 5880, 375,
285:Next i
    For i = 181 To 186: txt_5gn5(i - 1).Move 12480 + (i - 181) * ctr, 6960, 375,
285:Next i
    For i = 187 To 192: txt_5gn5(i - 1).Move 12480 + (i - 187) * ctr, 7320, 375,
285:Next i
    For i = 193 To 198: txt_5gn5(i - 1).Move 12480 + (i - 193) * ctr, 7680, 375,
285:Next i
    For i = 199 To 204: txt_5gn5(i - 1).Move 12480 + (i - 199) * ctr, 8040, 375,
285:Next i
    For i = 205 To 210: txt_5gn5(i - 1).Move 12480 + (i - 205) * ctr, 8400, 375,
285:Next i
    For i = 211 To 216: txt_5gn5(i - 1).Move 12480 + (i - 211) * ctr, 8760, 375,
285:Next i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll11.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll11.Left = ScaleWidth - VScroll11.Width
        If Picture1.Width > VScroll11.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll11.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll11.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll11.Left
    VScroll11.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll11.SmallChange = Abs(VScroll11.Max \ 16) + 1
    VScroll11.LargeChange = Abs(VScroll11.Max \ 4) + 1
    HScroll11.ZOrder 0
    VScroll11.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll11_Change()
    Picture1.Top = -VScroll11.Value
End Sub

Private Sub txt_5gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_5gn2.Text) > 6 Then
            resp = MsgBox("Maximum number of individuals in the second generation is 6", 0)
            txt_5gn2.SetFocus
        ElseIf Val(txt_5gn2.Text) <= 6 And Val(txt_5gn2.Text) <> 0 Then
            For i = 1 To Val(txt_5gn2.Text)
                txt_5gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_5gn3(i - 1).SetFocus
            Next i
            If Val(txt_5gn2.Text) > 1 And Val(txt_5gn2.Text) <= 3 Then VScroll11.Visible =
True
            If Val(txt_5gn2.Text) > 3 Then HScroll11.Visible = True: VScroll11.Visible = True
        End If
    End If
End Sub

Private Sub txt_5gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            If Val(txt_5gn3(i - 1).Text) > 6 Then
                resp = MsgBox("Maximum number of individuals in the third generation is 6", 0)
                txt_5gn3(i - 1).SetFocus
            ElseIf Val(txt_5gn3(i - 1).Text) <= 6 And Val(txt_5gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_5gn3(i - 1).Text)

```

```

        txt_5gn4(j - 1 + 6 * (i - 1)).Visible = True: Label5(j - 1 + 6 * (i -
1)).Visible = True
        If j = 1 Then txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
        Next j
    End If
Next i
End If
End Sub

Private Sub txt_5gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            For j = 1 To Val(txt_5gn3(i - 1).Text)
                If Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) > 6 Then
                    resp = MsgBox("Maximum number of individuals in the fourth generation is 6",
0)
                    txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                ElseIf Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) <= 6 And Val(txt_5gn4(j - 1 + 6
* (i - 1)).Text) <> 0 Then
                    For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                        txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True: Label4(j - 1
+ 6 * (i - 1)).Visible = True
                        If l = 1 Then txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).SetFocus
                        If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l =
Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.Enabled = True
                    Next l
                End If
            Next j
        Next i
    End If
End Sub

Private Sub txt_5gn5_KeyPress(Index As Integer, KeyAscii As Integer)
77 If KeyAscii = 9 Then
    For i = 1 To Val(txt_5gn2.Text)
        For j = 1 To Val(txt_5gn3(i - 1).Text)
            For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True
            Next l
            If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l =
Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.SetFocus
        Next j
    Next i
End If
End Sub

Private Sub cmd_back_Click()
    txt_5gn2.Text = Empty: txt_n5g.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 6: Label3(j - 1).Visible = False: txt_5gn3(j - 1).Text = Empty:
txt_5gn3(j - 1).Visible = False: Next j
    For i = 1 To 36
        Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
        txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
    Next i
    For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
    If tela = 5 Then tela = 0: frm_5g2.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_5g2.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 5 Then
            n5gi(2) = Val(txt_5gn2.Text)
            For j = 1 To n5gi(2)
                n5gij(2, j) = Val(txt_5gn3(j - 1).Text)
                For l = 1 To n5gij(2, j)
                    n5gijl(2, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                    For m = 1 To n5gijl(2, j, l)
                        n5gijlm(2, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                    Next m
                Next l
            Next j
            frm_screen.Visible = True: frm_5g2.Visible = False
        ElseIf tela = 6 Then

```

```

If Val(frm_cons1.txt_s25g.Text) <> 0 Then
    s25gi(x, 2) = Val(txt_5gn2.Text)
    For j = 1 To s25gi(x, 2)
        s25gij(x, 2, j) = Val(txt_5gn3(j - 1).Text)
        For l = 1 To s25gij(x, 2, j)
            s25gijl(x, 2, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
            For m = 1 To s25gijl(x, 2, j, l)
                s25gijlm(x, 2, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
            Next m
        Next l
    Next j
Else:
    s15gi(x, 2) = Val(txt_5gn2.Text)
    For j = 1 To s15gi(x, 2)
        s15gij(x, 2, j) = Val(txt_5gn3(j - 1).Text)
        For l = 1 To s15gij(x, 2, j)
            s15gijl(x, 2, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
            For m = 1 To s15gijl(x, 2, j, l)
                s15gijlm(x, 2, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
            Next m
        Next l
    Next j
    frm_cons1.Visible = True: frm_5g2.Visible = False
End If
End If
txt_5gn2.Text = Empty: cmd_enter.Enabled = False
For j = 1 To 6
    Label3(j - 1).Visible = False
    txt_5gn3(j - 1).Text = Empty: txt_5gn3(j - 1).Visible = False
Next j
For i = 1 To 36
    Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
    txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
Next i
For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
If Val(frm_5g1.txt_n5g.Text) > 2 Then
    frm_5g2.Visible = False: frm_5g3.Visible = True: frm_5g3.txt_5gn2.SetFocus
Else
    If tela = 5 Then tela = 0: frm_5g1.txt_n5g.Text = Empty: frm_5g2.Visible =
False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_5g1.txt_n5g.Text = Empty: frm_5g2.Visible =
False: frm_cons1.Visible = True
    End If
    ElseIf (resp = 7) Then frm_5g2.Visible = True
    End If
End Sub

```

FRM_5G3

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 2.2 * ScaleWidth, 2.5 * ScaleHeight
    txt_5gn2.Move 4680, 480, 375, 285
    txt_5gn3(0).Move 4680, 840, 375, 285: txt_5gn3(1).Move 4680, 3720, 375, 285
    txt_5gn3(2).Move 4680, 6600, 375, 285: txt_5gn3(3).Move 12120, 840, 375, 285
    txt_5gn3(4).Move 12120, 3720, 375, 285: txt_5gn3(5).Move 12120, 6600, 375, 285
    ctr = 360
    For i = 1 To 6
        txt_5gn4(i - 1).Move 3120, 1200 + (i - 1) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 1) * ctr, 1200, 375, 285
    Next i
    For i = 7 To 12
        txt_5gn4(i - 1).Move 3120, 4080 + (i - 7) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 7) * ctr, 1560, 375, 285
    Next i
    For i = 13 To 18
        txt_5gn4(i - 1).Move 3120, 6960 + (i - 13) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 13) * ctr, 1920, 375, 285
    Next i
    For i = 19 To 24
        txt_5gn4(i - 1).Move 10560, 1200 + (i - 19) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 19) * ctr, 2280, 375, 285
    Next i

```

```

For i = 25 To 30
    txt_5gn4(i - 1).Move 10560, 4080 + (i - 25) * ctr, 375, 285
    txt_5gn5(i - 1).Move 5040 + (i - 25) * ctr, 2640, 375, 285
Next i
For i = 31 To 36
    txt_5gn4(i - 1).Move 10560, 6960 + (i - 31) * ctr, 375, 285
    txt_5gn5(i - 1).Move 5040 + (i - 31) * ctr, 3000, 375, 285
Next i
For i = 37 To 42: txt_5gn5(i - 1).Move 5040 + (i - 37) * ctr, 4080, 375, 285: Next i
For i = 43 To 48: txt_5gn5(i - 1).Move 5040 + (i - 43) * ctr, 4440, 375, 285: Next i
For i = 49 To 54: txt_5gn5(i - 1).Move 5040 + (i - 49) * ctr, 4800, 375, 285: Next i
For i = 55 To 60: txt_5gn5(i - 1).Move 5040 + (i - 55) * ctr, 5160, 375, 285: Next i
For i = 61 To 66: txt_5gn5(i - 1).Move 5040 + (i - 61) * ctr, 5520, 375, 285: Next i
For i = 67 To 72: txt_5gn5(i - 1).Move 5040 + (i - 67) * ctr, 5880, 375, 285: Next i
For i = 73 To 78: txt_5gn5(i - 1).Move 5040 + (i - 73) * ctr, 6960, 375, 285: Next i
For i = 79 To 84: txt_5gn5(i - 1).Move 5040 + (i - 79) * ctr, 7320, 375, 285: Next i
For i = 85 To 90: txt_5gn5(i - 1).Move 5040 + (i - 85) * ctr, 7680, 375, 285: Next i
For i = 91 To 96: txt_5gn5(i - 1).Move 5040 + (i - 91) * ctr, 8040, 375, 285: Next i
For i = 97 To 102: txt_5gn5(i - 1).Move 5040 + (i - 97) * ctr, 8400, 375, 285: Next
i
    For i = 103 To 108: txt_5gn5(i - 1).Move 5040 + (i - 103) * ctr, 8760, 375, 285:
Next i
    For i = 109 To 114: txt_5gn5(i - 1).Move 12480 + (i - 109) * ctr, 1200, 375,
285:Next i
    For i = 115 To 120: txt_5gn5(i - 1).Move 12480 + (i - 115) * ctr, 1560, 375,
285:Next i
    For i = 121 To 126: txt_5gn5(i - 1).Move 12480 + (i - 121) * ctr, 1920, 375,
285:Next i
    For i = 127 To 132: txt_5gn5(i - 1).Move 12480 + (i - 127) * ctr, 2280, 375,
285:Next i
    For i = 133 To 138: txt_5gn5(i - 1).Move 12480 + (i - 133) * ctr, 2640, 375,
285:Next i
    For i = 139 To 144: txt_5gn5(i - 1).Move 12480 + (i - 139) * ctr, 3000, 375,
285:Next i
    For i = 145 To 150: txt_5gn5(i - 1).Move 12480 + (i - 145) * ctr, 4080, 375,
285:Next i
    For i = 151 To 156: txt_5gn5(i - 1).Move 12480 + (i - 151) * ctr, 4440, 375,
285:Next i
    For i = 157 To 162: txt_5gn5(i - 1).Move 12480 + (i - 157) * ctr, 4800, 375,
285:Next i
    For i = 163 To 168: txt_5gn5(i - 1).Move 12480 + (i - 163) * ctr, 5160, 375,
285:Next i
    For i = 169 To 174: txt_5gn5(i - 1).Move 12480 + (i - 169) * ctr, 5520, 375,
285:Next i
    For i = 175 To 180: txt_5gn5(i - 1).Move 12480 + (i - 175) * ctr, 5880, 375,
285:Next i
    For i = 181 To 186: txt_5gn5(i - 1).Move 12480 + (i - 181) * ctr, 6960, 375,
285:Next i
    For i = 187 To 192: txt_5gn5(i - 1).Move 12480 + (i - 187) * ctr, 7320, 375,
285:Next i
    For i = 193 To 198: txt_5gn5(i - 1).Move 12480 + (i - 193) * ctr, 7680, 375,
285:Next i
    For i = 199 To 204: txt_5gn5(i - 1).Move 12480 + (i - 199) * ctr, 8040, 375,
285:Next i
    For i = 205 To 210: txt_5gn5(i - 1).Move 12480 + (i - 205) * ctr, 8400, 375,
285:Next i
    For i = 211 To 216: txt_5gn5(i - 1).Move 12480 + (i - 211) * ctr, 8760, 375,
285:Next i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll11.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll11.Left = ScaleWidth - VScroll11.Width
        If Picture1.Width > VScroll11.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll11.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth

```

```

If HScroll1.Top > 0 Then VScroll1.Height = HScroll1.Top
HScroll1.Max = Picture1.Width - VScroll1.Left
VScroll1.Max = Picture1.Height - HScroll1.Top
HScroll1.SmallChange = Abs(HScroll1.Max \ 16) + 1
HScroll1.LargeChange = Abs(HScroll1.Max \ 4) + 1
VScroll1.SmallChange = Abs(VScroll1.Max \ 16) + 1
VScroll1.LargeChange = Abs(VScroll1.Max \ 4) + 1
HScroll1.ZOrder 0
VScroll1.ZOrder 0
End Sub

Sub HScroll1_Change()
    Picture1.Left = -HScroll1.Value
End Sub

Sub VScroll1_Change()
    Picture1.Top = -VScroll1.Value
End Sub

Private Sub txt_5gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_5gn2.Text) > 6 Then
            resp = MsgBox("Maximum number of individuals in the second generation is 6", 0)
            txt_5gn2.SetFocus
        ElseIf Val(txt_5gn2.Text) <= 6 And Val(txt_5gn2.Text) <> 0 Then
            For i = 1 To Val(txt_5gn2.Text)
                txt_5gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_5gn3(i - 1).SetFocus
            Next i
            If Val(txt_5gn2.Text) > 1 And Val(txt_5gn2.Text) <= 3 Then VScroll1.Visible =
True
            If Val(txt_5gn2.Text) > 3 Then HScroll1.Visible = True: VScroll1.Visible = True
            End If
        End If
    End Sub

Private Sub txt_5gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            If Val(txt_5gn3(i - 1).Text) > 6 Then
                resp = MsgBox("Maximum number of individuals in the third generation is 6", 0)
                txt_5gn3(i - 1).SetFocus
            ElseIf Val(txt_5gn3(i - 1).Text) <= 6 And Val(txt_5gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_5gn3(i - 1).Text)
                    txt_5gn4(j - 1 + 6 * (i - 1)).Visible = True: Label5(j - 1 + 6 * (i -
1)).Visible = True
                    If j = 1 Then txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                Next j
            End If
        Next i
    End Sub

Private Sub txt_5gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            For j = 1 To Val(txt_5gn3(i - 1).Text)
                If Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) > 6 Then
                    resp = MsgBox("Maximum number of individuals in the fourth generation is 6",
0)
                    txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                ElseIf Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) <= 6 And Val(txt_5gn4(j - 1 + 6
* (i - 1)).Text) <> 0 Then
                    For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                        txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True: Label4(j - 1
+ 6 * (i - 1)).Visible = True
                        If l = 1 Then txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).SetFocus
                        If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l =
Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.Enabled = True
                    Next l
                End If
            Next j
        Next i
    End Sub

Private Sub txt_5gn5_KeyPress(Index As Integer, KeyAscii As Integer)

```



```

77 If KeyAscii = 9 Then
    For i = 1 To Val(txt_5gn2.Text)
        For j = 1 To Val(txt_5gn3(i - 1).Text)
            For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True
            Next l
            If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l =
Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.SetFocus
        Next j
    Next i
End If
End Sub

Private Sub cmd_back_Click()
    txt_5gn2.Text = Empty: txt_n5g.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 6: Label3(j - 1).Visible = False: txt_5gn3(j - 1).Text = Empty:
txt_5gn3(j - 1).Visible = False: Next j
    For i = 1 To 36
        Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
        txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
    Next i
    For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
    If tela = 5 Then tela = 0: frm_5g3.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_5g3.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 5 Then
            n5gi(3) = Val(txt_5gn2.Text)
            For j = 1 To n5gi(3)
                n5gij(3, j) = Val(txt_5gn3(j - 1).Text)
                For l = 1 To n5gij(3, j)
                    n5gijl(3, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                    For m = 1 To n5gijl(3, j, l)
                        n5gijlm(3, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                    Next m
                Next l
            Next j
            frm_screen.Visible = True: frm_5g3.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s25g.Text) <> 0 Then
                s25gi(x, 3) = Val(txt_5gn2.Text)
                For j = 1 To s25gi(x, 3)
                    s25gij(x, 3, j) = Val(txt_5gn3(j - 1).Text)
                    For l = 1 To s25gij(x, 3, j)
                        s25gijl(x, 3, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                        For m = 1 To s25gijl(x, 3, j, l)
                            s25gijlm(x, 3, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                        Next m
                    Next l
                Next j
            Else:
                s15gi(x, 3) = Val(txt_5gn2.Text)
                For j = 1 To s15gi(x, 3)
                    s15gij(x, 3, j) = Val(txt_5gn3(j - 1).Text)
                    For l = 1 To s15gij(x, 3, j)
                        s15gijl(x, 3, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                        For m = 1 To s15gijl(x, 3, j, l)
                            s15gijlm(x, 3, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                        Next m
                    Next l
                Next j
                frm_cons1.Visible = True: frm_5g3.Visible = False
            End If
        End If
        'limpa as caixas de texto
        txt_5gn2.Text = Empty: cmd_enter.Enabled = False
        For j = 1 To 6
            Label3(j - 1).Visible = False
            txt_5gn3(j - 1).Text = Empty: txt_5gn3(j - 1).Visible = False
        Next j
    End If

```

```

For i = 1 To 36
    Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
    txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
Next i
For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
If Val(frm_5g1.txt_n5g.Text) > 3 Then
    frm_5g3.Visible = False: frm_5g4.Visible = True: frm_5g4.txt_5gn2.SetFocus
Else
    If tela = 5 Then tela = 0: frm_5g1.txt_n5g.Text = Empty: frm_5g3.Visible =
False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_5g1.txt_n5g.Text = Empty: frm_5g3.Visible =
False: frm_consl.Visible = True
    End If
    ElseIf (resp = 7) Then frm_5g3.Visible = True
    End If
End Sub

```

FRM_5G4

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 2.2 * ScaleWidth, 2.5 * ScaleHeight
    txt_5gn2.Move 4680, 480, 375, 285
    txt_5gn3(0).Move 4680, 840, 375, 285: txt_5gn3(1).Move 4680, 3720, 375, 285
    txt_5gn3(2).Move 4680, 6600, 375, 285: txt_5gn3(3).Move 12120, 840, 375, 285
    txt_5gn3(4).Move 12120, 3720, 375, 285: txt_5gn3(5).Move 12120, 6600, 375, 285
    ctr = 360
    For i = 1 To 6
        txt_5gn4(i - 1).Move 3120, 1200 + (i - 1) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 1) * ctr, 1200, 375, 285
    Next i
    For i = 7 To 12
        txt_5gn4(i - 1).Move 3120, 4080 + (i - 7) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 7) * ctr, 1560, 375, 285
    Next i
    For i = 13 To 18
        txt_5gn4(i - 1).Move 3120, 6960 + (i - 13) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 13) * ctr, 1920, 375, 285
    Next i
    For i = 19 To 24
        txt_5gn4(i - 1).Move 10560, 1200 + (i - 19) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 19) * ctr, 2280, 375, 285
    Next i
    For i = 25 To 30
        txt_5gn4(i - 1).Move 10560, 4080 + (i - 25) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 25) * ctr, 2640, 375, 285
    Next i
    For i = 31 To 36
        txt_5gn4(i - 1).Move 10560, 6960 + (i - 31) * ctr, 375, 285
        txt_5gn5(i - 1).Move 5040 + (i - 31) * ctr, 3000, 375, 285
    Next i
    For i = 37 To 42: txt_5gn5(i - 1).Move 5040 + (i - 37) * ctr, 4080, 375, 285: Next i
    For i = 43 To 48: txt_5gn5(i - 1).Move 5040 + (i - 43) * ctr, 4440, 375, 285: Next i
    For i = 49 To 54: txt_5gn5(i - 1).Move 5040 + (i - 49) * ctr, 4800, 375, 285: Next i
    For i = 55 To 60: txt_5gn5(i - 1).Move 5040 + (i - 55) * ctr, 5160, 375, 285: Next i
    For i = 61 To 66: txt_5gn5(i - 1).Move 5040 + (i - 61) * ctr, 5520, 375, 285: Next i
    For i = 67 To 72: txt_5gn5(i - 1).Move 5040 + (i - 67) * ctr, 5880, 375, 285: Next i

    For i = 73 To 78: txt_5gn5(i - 1).Move 5040 + (i - 73) * ctr, 6960, 375, 285: Next i
    For i = 79 To 84: txt_5gn5(i - 1).Move 5040 + (i - 79) * ctr, 7320, 375, 285: Next i
    For i = 85 To 90: txt_5gn5(i - 1).Move 5040 + (i - 85) * ctr, 7680, 375, 285: Next i
    For i = 91 To 96: txt_5gn5(i - 1).Move 5040 + (i - 91) * ctr, 8040, 375, 285: Next i
    For i = 97 To 102: txt_5gn5(i - 1).Move 5040 + (i - 97) * ctr, 8400, 375, 285: Next
i
    For i = 103 To 108: txt_5gn5(i - 1).Move 5040 + (i - 103) * ctr, 8760, 375, 285:
Next i
    For i = 109 To 114: txt_5gn5(i - 1).Move 12480 + (i - 109) * ctr, 1200, 375,
285:Next i
    For i = 115 To 120: txt_5gn5(i - 1).Move 12480 + (i - 115) * ctr, 1560, 375,
285:Next i
    For i = 121 To 126: txt_5gn5(i - 1).Move 12480 + (i - 121) * ctr, 1920, 375,
285:Next i
    For i = 127 To 132: txt_5gn5(i - 1).Move 12480 + (i - 127) * ctr, 2280, 375,
285:Next i

```

```

    For i = 133 To 138: txt_5gn5(i - 1).Move 12480 + (i - 133) * ctr, 2640, 375,
285:Next i
    For i = 139 To 144: txt_5gn5(i - 1).Move 12480 + (i - 139) * ctr, 3000, 375,
285:Next i
    For i = 145 To 150: txt_5gn5(i - 1).Move 12480 + (i - 145) * ctr, 4080, 375,
285:Next i
    For i = 151 To 156: txt_5gn5(i - 1).Move 12480 + (i - 151) * ctr, 4440, 375,
285:Next i
    For i = 157 To 162: txt_5gn5(i - 1).Move 12480 + (i - 157) * ctr, 4800, 375,
285:Next i
    For i = 163 To 168: txt_5gn5(i - 1).Move 12480 + (i - 163) * ctr, 5160, 375,
285:Next i
    For i = 169 To 174: txt_5gn5(i - 1).Move 12480 + (i - 169) * ctr, 5520, 375,
285:Next i
    For i = 175 To 180: txt_5gn5(i - 1).Move 12480 + (i - 175) * ctr, 5880, 375,
285:Next i
    For i = 181 To 186: txt_5gn5(i - 1).Move 12480 + (i - 181) * ctr, 6960, 375,
285:Next i
    For i = 187 To 192: txt_5gn5(i - 1).Move 12480 + (i - 187) * ctr, 7320, 375,
285:Next i
    For i = 193 To 198: txt_5gn5(i - 1).Move 12480 + (i - 193) * ctr, 7680, 375,
285:Next i
    For i = 199 To 204: txt_5gn5(i - 1).Move 12480 + (i - 199) * ctr, 8040, 375,
285:Next i
    For i = 205 To 210: txt_5gn5(i - 1).Move 12480 + (i - 205) * ctr, 8400, 375,
285:Next i
    For i = 211 To 216: txt_5gn5(i - 1).Move 12480 + (i - 211) * ctr, 8760, 375,
285:Next i
End Sub

Sub Form_Resize()
    HScroll11.Left = 0
    VScroll11.Top = 0
    If Picture1.Width > ScaleWidth Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    Else
        HScroll11.Top = ScaleHeight
    End If
    If Picture1.Height > HScroll11.Top Then
        VScroll11.Left = ScaleWidth - VScroll11.Width
        If Picture1.Width > VScroll11.Left Then
            HScroll11.Top = ScaleHeight - HScroll11.Height
        End If
    Else
        VScroll11.Left = ScaleWidth
    End If
    HScroll11.Width = ScaleWidth
    If HScroll11.Top > 0 Then VScroll11.Height = HScroll11.Top
    HScroll11.Max = Picture1.Width - VScroll11.Left
    VScroll11.Max = Picture1.Height - HScroll11.Top
    HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
    HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
    VScroll11.SmallChange = Abs(VScroll11.Max \ 16) + 1
    VScroll11.LargeChange = Abs(VScroll11.Max \ 4) + 1
    HScroll11.ZOrder 0
    VScroll11.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll11_Change()
    Picture1.Top = -VScroll11.Value
End Sub

Private Sub txt_5gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_5gn2.Text) > 6 Then
            resp = MsgBox("Maximum number of individuals in the second generation is 6", 0)
            txt_5gn2.SetFocus
        ElseIf Val(txt_5gn2.Text) <= 6 And Val(txt_5gn2.Text) <> 0 Then
            For i = 1 To Val(txt_5gn2.Text)
                txt_5gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_5gn3(i - 1).SetFocus
            Next i
        End If
    End If
End Sub

```

```

        If Val(txt_5gn2.Text) > 1 And Val(txt_5gn2.Text) <= 3 Then VScroll1.Visible =
True
        If Val(txt_5gn2.Text) > 3 Then HScroll1.Visible = True: VScroll1.Visible = True
    End If
End If
End Sub

Private Sub txt_5gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            If Val(txt_5gn3(i - 1).Text) > 6 Then
                resp = MsgBox("Maximum number of individuals in the third generation is 6", 0)
                txt_5gn3(i - 1).SetFocus
            ElseIf Val(txt_5gn3(i - 1).Text) <= 6 And Val(txt_5gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_5gn3(i - 1).Text)
                    txt_5gn4(j - 1 + 6 * (i - 1)).Visible = True: Label5(j - 1 + 6 * (i -
1)).Visible = True
                    If j = 1 Then txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                Next j
            End If
        Next i
    End If
End Sub

Private Sub txt_5gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            For j = 1 To Val(txt_5gn3(i - 1).Text)
                If Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) > 6 Then
                    resp = MsgBox("Maximum number of individuals in the fourth generation is 6",
0)
                    txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                ElseIf Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) <= 6 And Val(txt_5gn4(j - 1 + 6
* (i - 1)).Text) <> 0 Then
                    For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                        txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True: Label4(j - 1
+ 6 * (i - 1)).Visible = True
                        If l = 1 Then txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).SetFocus
                        If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l =
Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.Enabled = True
                    Next l
                End If
            Next j
        Next i
    End If
End Sub

Private Sub txt_5gn5_KeyPress(Index As Integer, KeyAscii As Integer)
77 If KeyAscii = 9 Then
    For i = 1 To Val(txt_5gn2.Text)
        For j = 1 To Val(txt_5gn3(i - 1).Text)
            For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True
            Next l
            If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l =
Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.SetFocus
        Next j
    Next i
End If
End Sub

Private Sub cmd_back_Click()
    txt_5gn2.Text = Empty: txt_n5g.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 6: Label3(j - 1).Visible = False: txt_5gn3(j - 1).Text = Empty:
txt_5gn3(j - 1).Visible = False: Next j
    For i = 1 To 36
        Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
        txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
    Next i
    For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
    If tela = 5 Then tela = 0: frm_5g4.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_5g4.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)

```

```

If (resp = 6) Then
    If tela = 5 Then
        n5gi(4) = Val(txt_5gn2.Text)
        For j = 1 To n5gi(4)
            n5gij(4, j) = Val(txt_5gn3(j - 1).Text)
            For l = 1 To n5gij(4, j)
                n5gijl(4, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                For m = 1 To n5gijl(4, j, l)
                    n5gijlm(4, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                Next m
            Next l
        Next j
        frm_screen.Visible = True: frm_5g4.Visible = False
    ElseIf tela = 6 Then
        If Val(frm_cons1.txt_s25g.Text) <> 0 Then
            s25gi(x, 4) = Val(txt_5gn2.Text)
            For j = 1 To s25gi(x, 4)
                s25gij(x, 4, j) = Val(txt_5gn3(j - 1).Text)
                For l = 1 To s25gij(x, 4, j)
                    s25gijl(x, 4, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                    For m = 1 To s25gijl(x, 4, j, l)
                        s25gijlm(x, 4, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                    Next m
                Next l
            Next j
        Else:
            s15gi(x, 4) = Val(txt_5gn2.Text)
            For j = 1 To s15gi(x, 4)
                s15gij(x, 4, j) = Val(txt_5gn3(j - 1).Text)
                For l = 1 To s15gij(x, 4, j)
                    s15gijl(x, 4, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                    For m = 1 To s15gijl(x, 4, j, l)
                        s15gijlm(x, 4, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                    Next m
                Next l
            Next j
            frm_cons1.Visible = True: frm_5g4.Visible = False
        End If
    End If
    'limpa as caixas de texto
    txt_5gn2.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 6
        Label3(j - 1).Visible = False
        txt_5gn3(j - 1).Text = Empty: txt_5gn3(j - 1).Visible = False
    Next j
    For i = 1 To 36
        Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
        txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
    Next i
    For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
    If Val(frm_5g1.txt_n5g.Text) > 4 Then
        frm_5g4.Visible = False: frm_5g5.Visible = True: frm_5g5.txt_5gn2.SetFocus
    Else
        If tela = 5 Then tela = 0: frm_5g1.txt_n5g.Text = Empty: frm_5g4.Visible =
False: frm_screen.Visible = True
        If tela = 6 Then tela = 0: frm_5g1.txt_n5g.Text = Empty: frm_5g4.Visible =
False: frm_cons1.Visible = True
    End If
    ElseIf (resp = 7) Then frm_5g4.Visible = True
    End If
End Sub

```

FRM_5G5

```

Private Sub Form_Load()
    Picture1.Move 120, 240, 2.2 * ScaleWidth, 2.5 * ScaleHeight
    txt_5gn2.Move 4680, 480, 375, 285
    txt_5gn3(0).Move 4680, 840, 375, 285: txt_5gn3(1).Move 4680, 3720, 375, 285
    txt_5gn3(2).Move 4680, 6600, 375, 285: txt_5gn3(3).Move 12120, 840, 375, 285
    txt_5gn3(4).Move 12120, 3720, 375, 285: txt_5gn3(5).Move 12120, 6600, 375, 285
    ctr = 360

```

```

For i = 1 To 6
    txt_5gn4(i - 1).Move 3120, 1200 + (i - 1) * ctr, 375, 285
    txt_5gn5(i - 1).Move 5040 + (i - 1) * ctr, 1200, 375, 285
Next i
For i = 7 To 12
    txt_5gn4(i - 1).Move 3120, 4080 + (i - 7) * ctr, 375, 285
    txt_5gn5(i - 1).Move 5040 + (i - 7) * ctr, 1560, 375, 285
Next i
For i = 13 To 18
    txt_5gn4(i - 1).Move 3120, 6960 + (i - 13) * ctr, 375, 285
    txt_5gn5(i - 1).Move 5040 + (i - 13) * ctr, 1920, 375, 285
Next i
For i = 19 To 24
    txt_5gn4(i - 1).Move 10560, 1200 + (i - 19) * ctr, 375, 285
    txt_5gn5(i - 1).Move 5040 + (i - 19) * ctr, 2280, 375, 285
Next i
For i = 25 To 30
    txt_5gn4(i - 1).Move 10560, 4080 + (i - 25) * ctr, 375, 285
    txt_5gn5(i - 1).Move 5040 + (i - 25) * ctr, 2640, 375, 285
Next i
For i = 31 To 36
    txt_5gn4(i - 1).Move 10560, 6960 + (i - 31) * ctr, 375, 285
    txt_5gn5(i - 1).Move 5040 + (i - 31) * ctr, 3000, 375, 285
Next i
For i = 37 To 42: txt_5gn5(i - 1).Move 5040 + (i - 37) * ctr, 4080, 375, 285: Next i
For i = 43 To 48: txt_5gn5(i - 1).Move 5040 + (i - 43) * ctr, 4440, 375, 285: Next i
For i = 49 To 54: txt_5gn5(i - 1).Move 5040 + (i - 49) * ctr, 4800, 375, 285: Next i
For i = 55 To 60: txt_5gn5(i - 1).Move 5040 + (i - 55) * ctr, 5160, 375, 285: Next i
For i = 61 To 66: txt_5gn5(i - 1).Move 5040 + (i - 61) * ctr, 5520, 375, 285: Next i
For i = 67 To 72: txt_5gn5(i - 1).Move 5040 + (i - 67) * ctr, 5880, 375, 285: Next i
For i = 73 To 78: txt_5gn5(i - 1).Move 5040 + (i - 73) * ctr, 6960, 375, 285: Next i
For i = 79 To 84: txt_5gn5(i - 1).Move 5040 + (i - 79) * ctr, 7320, 375, 285: Next i
For i = 85 To 90: txt_5gn5(i - 1).Move 5040 + (i - 85) * ctr, 7680, 375, 285: Next i
For i = 91 To 96: txt_5gn5(i - 1).Move 5040 + (i - 91) * ctr, 8040, 375, 285: Next i
For i = 97 To 102: txt_5gn5(i - 1).Move 5040 + (i - 97) * ctr, 8400, 375, 285: Next
i
For i = 103 To 108: txt_5gn5(i - 1).Move 5040 + (i - 103) * ctr, 8760, 375, 285:
Next i
For i = 109 To 114: txt_5gn5(i - 1).Move 12480 + (i - 109) * ctr, 1200, 375,
285:Next i
For i = 115 To 120: txt_5gn5(i - 1).Move 12480 + (i - 115) * ctr, 1560, 375,
285:Next i
For i = 121 To 126: txt_5gn5(i - 1).Move 12480 + (i - 121) * ctr, 1920, 375,
285:Next i
For i = 127 To 132: txt_5gn5(i - 1).Move 12480 + (i - 127) * ctr, 2280, 375,
285:Next i
For i = 133 To 138: txt_5gn5(i - 1).Move 12480 + (i - 133) * ctr, 2640, 375,
285:Next i
For i = 139 To 144: txt_5gn5(i - 1).Move 12480 + (i - 139) * ctr, 3000, 375,
285:Next i
For i = 145 To 150: txt_5gn5(i - 1).Move 12480 + (i - 145) * ctr, 4080, 375,
285:Next i
For i = 151 To 156: txt_5gn5(i - 1).Move 12480 + (i - 151) * ctr, 4440, 375,
285:Next i
For i = 157 To 162: txt_5gn5(i - 1).Move 12480 + (i - 157) * ctr, 4800, 375,
285:Next i
For i = 163 To 168: txt_5gn5(i - 1).Move 12480 + (i - 163) * ctr, 5160, 375,
285:Next i
For i = 169 To 174: txt_5gn5(i - 1).Move 12480 + (i - 169) * ctr, 5520, 375,
285:Next i
For i = 175 To 180: txt_5gn5(i - 1).Move 12480 + (i - 175) * ctr, 5880, 375,
285:Next i
For i = 181 To 186: txt_5gn5(i - 1).Move 12480 + (i - 181) * ctr, 6960, 375,
285:Next i
For i = 187 To 192: txt_5gn5(i - 1).Move 12480 + (i - 187) * ctr, 7320, 375,
285:Next i
For i = 193 To 198: txt_5gn5(i - 1).Move 12480 + (i - 193) * ctr, 7680, 375,
285:Next i
For i = 199 To 204: txt_5gn5(i - 1).Move 12480 + (i - 199) * ctr, 8040, 375,
285:Next i
For i = 205 To 210: txt_5gn5(i - 1).Move 12480 + (i - 205) * ctr, 8400, 375,
285:Next i
For i = 211 To 216: txt_5gn5(i - 1).Move 12480 + (i - 211) * ctr, 8760, 375,
285:Next i
End Sub

Sub Form_Resize()

```

```

HScroll11.Left = 0
VScroll11.Top = 0
If Picture1.Width > ScaleWidth Then
    HScroll11.Top = ScaleHeight - HScroll11.Height
Else
    HScroll11.Top = ScaleHeight
End If
If Picture1.Height > HScroll11.Top Then
    VScroll11.Left = ScaleWidth - VScroll11.Width
    If Picture1.Width > VScroll11.Left Then
        HScroll11.Top = ScaleHeight - HScroll11.Height
    End If
Else
    VScroll11.Left = ScaleWidth
End If
HScroll11.Width = ScaleWidth
If HScroll11.Top > 0 Then VScroll11.Height = HScroll11.Top
HScroll11.Max = Picture1.Width - VScroll11.Left
VScroll11.Max = Picture1.Height - HScroll11.Top
HScroll11.SmallChange = Abs(HScroll11.Max \ 16) + 1
HScroll11.LargeChange = Abs(HScroll11.Max \ 4) + 1
VScroll11.SmallChange = Abs(VScroll11.Max \ 16) + 1
VScroll11.LargeChange = Abs(VScroll11.Max \ 4) + 1
HScroll11.ZOrder 0
VScroll11.ZOrder 0
End Sub

Sub HScroll11_Change()
    Picture1.Left = -HScroll11.Value
End Sub

Sub VScroll11_Change()
    Picture1.Top = -VScroll11.Value
End Sub

Private Sub txt_5gn2_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Val(txt_5gn2.Text) > 6 Then
            resp = MsgBox("Maximum number of individuals in the second generation is 6", 0)
            txt_5gn2.SetFocus
        ElseIf Val(txt_5gn2.Text) <= 6 And Val(txt_5gn2.Text) <> 0 Then
            For i = 1 To Val(txt_5gn2.Text)
                txt_5gn3(i - 1).Visible = True: Label3(i - 1).Visible = True
                If i = 1 Then txt_5gn3(i - 1).SetFocus
            Next i
            If Val(txt_5gn2.Text) > 1 And Val(txt_5gn2.Text) <= 3 Then VScroll11.Visible =
True
                If Val(txt_5gn2.Text) > 3 Then HScroll11.Visible = True: VScroll11.Visible = True
            End If
        End If
    End Sub

Private Sub txt_5gn3_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            If Val(txt_5gn3(i - 1).Text) > 6 Then
                resp = MsgBox("Maximum number of individuals in the third generation is 6", 0)
                txt_5gn3(i - 1).SetFocus
            ElseIf Val(txt_5gn3(i - 1).Text) <= 6 And Val(txt_5gn3(i - 1).Text) <> 0 Then
                For j = 1 To Val(txt_5gn3(i - 1).Text)
                    txt_5gn4(j - 1 + 6 * (i - 1)).Visible = True: Label5(j - 1 + 6 * (i -
1)).Visible = True
                    If j = 1 Then txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                Next j
            End If
        Next i
    End Sub

Private Sub txt_5gn4_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        For i = 1 To Val(txt_5gn2.Text)
            For j = 1 To Val(txt_5gn3(i - 1).Text)
                If Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) > 6 Then
                    resp = MsgBox("Maximum number of individuals in the fourth generation is 6",
0)
                    txt_5gn4(j - 1 + 6 * (i - 1)).SetFocus
                End If
            Next j
        Next i
    End Sub

```

```

ElseIf Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) <= 6 And Val(txt_5gn4(j - 1 + 6
* (i - 1)).Text) <> 0 Then
    For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
        txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True: Label4(j - 1
+ 6 * (i - 1)).Visible = True
        If l = 1 Then txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).SetFocus
        If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l =
Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.Enabled = True
    Next l
End If
Next j
Next i
End If
End Sub

Private Sub txt_5gn5_KeyPress(Index As Integer, KeyAscii As Integer)
77 If KeyAscii = 9 Then
    For i = 1 To Val(txt_5gn2.Text)
        For j = 1 To Val(txt_5gn3(i - 1).Text)
            For l = 1 To Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text)
                txt_5gn5(l - 1 + 6 * (j - 1) + 36 * (i - 1)).Visible = True
            Next l
            If i = Val(txt_5gn2.Text) And j = Val(txt_5gn3(i - 1).Text) And l =
Val(txt_5gn4(j - 1 + 6 * (i - 1)).Text) Then cmd_enter.SetFocus
        Next j
    Next i
End If
End Sub

Private Sub cmd_back_Click()
    txt_5gn2.Text = Empty: txt_n5g.Text = Empty: cmd_enter.Enabled = False
    For j = 1 To 6: Label3(j - 1).Visible = False: txt_5gn3(j - 1).Text = Empty:
txt_5gn3(j - 1).Visible = False: Next j
    For i = 1 To 36
        Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
        txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
    Next i
    For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
    If tela = 5 Then tela = 0: frm_5g5.Visible = False: frm_screen.Visible = True
    If tela = 6 Then tela = 0: frm_5g5.Visible = False: frm_cons1.Visible = True
End Sub

Private Sub cmd_enter_Click()
    resp = MsgBox("Are the data correct?", 4)
    If (resp = 6) Then
        If tela = 5 Then
            n5gi(5) = Val(txt_5gn2.Text)
            For j = 1 To n5gi(5)
                n5gij(5, j) = Val(txt_5gn3(j - 1).Text)
                For l = 1 To n5gij(5, j)
                    n5gijl(5, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                    For m = 1 To n5gijl(5, j, l)
                        n5gijlm(5, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                    Next m
                Next l
            Next j
            frm_screen.Visible = True: frm_5g5.Visible = False
        ElseIf tela = 6 Then
            If Val(frm_cons1.txt_s25g.Text) <> 0 Then
                s25gi(x, 5) = Val(txt_5gn2.Text)
                For j = 1 To s25gi(x, 5)
                    s25gij(x, 5, j) = Val(txt_5gn3(j - 1).Text)
                    For l = 1 To s25gij(x, 5, j)
                        s25gijl(x, 5, j, l) = Val(txt_5gn4(l - 1 + 6 * (j - 1)).Text)
                        For m = 1 To s25gijl(x, 5, j, l)
                            s25gijlm(x, 5, j, l, m) = Val(txt_5gn5(m - 1 + 6 * (l - 1) + 36 * (j -
1)).Text)
                        Next m
                    Next l
                Next j
            Else:
                s15gi(x, 5) = Val(txt_5gn2.Text)
                For j = 1 To s15gi(x, 5)
                    s15gij(x, 5, j) = Val(txt_5gn3(j - 1).Text)
                    For l = 1 To s15gij(x, 5, j)

```



```

s15gijl(x, 5, j, 1) = Val(txt_5gn4(1 - 1 + 6 * (j - 1)).Text)
For m = 1 To s15gijl(x, 5, j, 1)
    s15gijlm(x, 5, j, 1, m) = Val(txt_5gn5(m - 1 + 6 * (1 - 1) + 36 * (j -
1)).Text)
    Next m
Next l
Next j
frm_cons1.Visible = True: frm_5g5.Visible = False
End If
End If
'limpa as caixas de texto
txt_5gn2.Text = Empty: cmd_enter.Enabled = False
For j = 1 To 6
    Label3(j - 1).Visible = False
    txt_5gn3(j - 1).Text = Empty: txt_5gn3(j - 1).Visible = False
Next j
For i = 1 To 36
    Label4(i - 1).Visible = False: Label5(i - 1).Visible = False
    txt_5gn4(i - 1).Text = Empty: txt_5gn4(i - 1).Visible = False
Next i
For i = 1 To 216: txt_5gn5(i - 1).Text = Empty: txt_5gn5(i - 1).Visible = False:
Next i
If tela = 5 Then tela = 0: frm_5g1.txt_n5g.Text = Empty: frm_5g5.Visible = False:
frm_screen.Visible = True
If tela = 6 Then tela = 0: frm_5g1.txt_n5g.Text = Empty: frm_5g5.Visible = False:
frm_cons1.Visible = True
ElseIf (resp = 7) Then frm_5g5.Visible = True
End If
End Sub

```