

```

1: #include <stdlib.h>
2: #include <stdio.h>
3: #include <string.h>
4: #include <math.h>
5:
6: void imprime(float m[3][7], int linhas, int colunas, char l[3][3], int tipo){
7:     int i, j;
8:     printf ("      | x1\t\tx2\t\tf1\t\tf2\t\tta1\t\tta2\t\t\ttcoef");
9:     printf("\n");
10:    for ( i=0; i<linhas; i++ ){
11:        printf ("%s | ", l[i]);
12:        for ( j = 0; j<colunas; j++ ){
13:            if(tipo == 1){
14:                printf("%.2f\t\t", m[i][j]);
15:            }else{
16:                printf("%.2e\t", m[i][j]);
17:            }
18:        }
19:        printf("\n");
20:    }
21:    printf("\n\n");
22: }
23:
24: int main(){
25:
26:     float simplex[3][7], totfert, totdef, ferta, fertb, defa, defb, precoa, precob;
27:     float x1, x2, f1, f2, a1, a2, bigM, pp[2], menor, divisor0, divisor1, divisor2;
28:     int i, j, k, tipoMatriz, colunaMenor;
29:     char continua, futuroNome[3];
30:     char nomeLinha[3][3];
31:
32:
33:     printf ("\nUma empresa de agronegocio usa avioes para pulverizar defensivos
agricolas e fertilizantes.");
34:     printf ("\nExistem dois modelos distintos de avioes que possuem dois tanques
cada.");
35:     printf ("\nAvioes das marcas A e B com capacidades especificas dos tanques de
defensivos e fertilizantes.");
36:     printf ("\nDefensivos e fertilizantes sao pulverizados ao mesmo tempo em uma
mesma viagem.");
37:     printf ("\nCada aviao possui um custo especifico por viagem.");
38:     printf ("\nA empresa deseja pulverizar quantidades determinadas de defensivos e
de fertilizantes a um custo minimo.\n\n");
39:
40:     printf ("\nAvioes da marca A : tanques de defensivos -> 600 litros; Tanque de
fertilizantes -> 900 litros;");
41:     printf ("\nAvioes da marca B : tanques de defensivos -> 600 litros; Tanque de
fertilizantes -> 300 litros;");
42:     printf ("\nE necessario pulverizar 4800 litros de defensivos e 3600 litros de
fertilizantes.");
43:     printf ("\nCusto de cada voo dos avioes A e B:");
44:     printf ("\nAvioes da marca A : R$ 1650,00");
45:     printf ("\nAvioes da marca B : R$ 1125,00");
46:
47:     printf ("\nQuantos voos de cada um dos avioes serao necessarios para minimizar");

```

```

48:     printf ("\na funcao custo e executar a tarefa desejada pela empresa de
agronegocio?\n\n");
49:
50:
51:     printf ("\nDeseja modificar os valores acima?(s/n)\n");
52:     scanf("%c", &continua);
53:     if(continua=='s' || continua=='S'){
54:         printf ("\nInforme o total de defensivos a ser pulverizado:");
55:         scanf("%f", &totdef);
56:         printf ("\nInforme o total de fertilizantes a ser pulverizado:");
57:         scanf("%f", &totfert);
58:
59:         printf ("\nInforme a capacidade do tanque de defensivos do aviao A:");
60:         scanf("%f", &defa);
61:         printf ("\nInforme a capacidade do tanque de fertilizantes do aviao A:");
62:         scanf("%f", &ferta);
63:
64:         printf ("\nInforme a capacidade do tanque de defensivos do aviao B:");
65:         scanf("%f", &defb);
66:         printf ("\nInforme a capacidade do tanque de fertilizantes do aviao B:");
67:         scanf("%f", &fertb);
68:
69:         printf ("\nInforme o custo total de cada viagem do aviao A:");
70:         scanf("%f", &precoa);
71:         printf ("\nInforme o custo total de cada viagem do aviao B:");
72:         scanf("%f", &precob);
73:     }else{
74:         totdef = 4800;
75:         totfert = 3600;
76:         defa = 600;
77:         ferta = 900;
78:         defb = 600;
79:         fertb = 300;
80:         precoa = 1650;
81:         precob = 1125;
82:     }
83:
84:     printf("\nVizualizacao da matriz : 1 para matriz com decimais OU 2 para matriz
com notacao cientifica.\n");
85:     scanf("%d", &tipoMatriz);
86:
87:     printf("\nA seguir, as funcoes que modelam o problema : \n");
88:
89:     printf("\nZmin = %.1f X1 + %.1f X2\n", precoa, precob);
90:     printf("\n%.1f X1 + %.1f X2 >= %.1f", defa, defb, totdef);
91:     printf("\n%.1f X1 + %.1f X2 >= %.1f\n", ferta, fertb, totfert);
92:     printf("\nO M adotado para a resolucao do metodo sera sempre a soma dos precos
dos avioes A e B.");
93:     printf("\nQue neste caso resultara em %.2f.\n\n", precoa+precob);
94:
95:     // x1 e x2 são a quantidade de viagens que cada um dos aviões a e b farão,
    respectivamente.
96:
97:     //Construção da matriz:
98:     k = 1;

```

```

99:
100: strcpy(nomeLinha[0], "a1");
101: strcpy(nomeLinha[1], "a2");
102: strcpy(nomeLinha[2], "Z ");
103:
104: bigM = (precoa + precob);
105:
106: simplex[0][0] = defa;
107: simplex[0][1] = defb;
108: simplex[0][2] = -1;
109: simplex[0][3] = 0;
110: simplex[0][4] = 1;
111: simplex[0][5] = 0;
112: simplex[0][6] = totdef;
113: simplex[1][0] = ferta;
114: simplex[1][1] = fertb;
115: simplex[1][2] = 0;
116: simplex[1][3] = -1;
117: simplex[1][4] = 0;
118: simplex[1][5] = 1;
119: simplex[1][6] = totfert;
120: simplex[2][0] = precoa;
121: simplex[2][1] = precob;
122: simplex[2][2] = 0;
123: simplex[2][3] = 0;
124: simplex[2][4] = bigM;
125: simplex[2][5] = bigM;
126: simplex[2][6] = 0;
127:
128: imprime(simplex, 3, 7, nomeLinha, tipoMatriz);
129:
130: printf("Tirar o M das colunas das variaveis artificiais na linha do Z.\n\n");
131:
132: //Tirar o M das colunas das variaveis artificiais na linha do Z
133: for(i=0; i<7; i++){
134:     simplex[2][i] = simplex[2][i] - (simplex[1][i])*bigM;
135:     simplex[2][i] = simplex[2][i] - (simplex[0][i])*bigM;
136: }
137:
138: imprime(simplex, 3, 7, nomeLinha, tipoMatriz);
139:
140: //Início de um ciclo aqui!!!
141: while (1){
142: //Verifica se já não há mais variáveis artificiais na solução
143:     if(strcmp(nomeLinha[0], "a1") != 0 && strcmp(nomeLinha[1], "a2") != 0){
144:         for(i=0; i<4; i++){
145:             if(simplex[2][i] < 0){
146:                 break;
147:             }
148:         }
149:         if(i == 4){
150: //Com i igual a 4, obtemos o quadro ótimo, pois não há coeficiente negativo em
nenhuma variavel não artificial da linha Z.
151:             break;
152:         }

```

```

153:     }
154: //Fazer o processo de produção para cada linha
155:     menor = bigM;
156:     for(i=0;i<4;i++){
157:         if(menor > simplex[2][i]){
158:             menor = simplex[2][i];
159:             colunaMenor = i;
160:         }
161:     }
162:
163: //A seguir será atribuído o nome da variável que tomará uma posição na coluna da
    esquerda, inicialmente contendo as variáveis artificiais
164:     switch (colunaMenor){
165:         case 0:
166:             strcpy(futuroNome, "x1");
167:             break;
168:         case 1:
169:             strcpy(futuroNome, "x2");
170:             break;
171:         case 2:
172:             strcpy(futuroNome, "f1");
173:             break;
174:         case 3:
175:             strcpy(futuroNome, "f2");
176:             break;
177:     }
178:
179:     printf("\n Coluna menor : %d\t\t Iteracao numero : %d\n\n", colunaMenor,
    k++);
180:
181: //Obtendo o pp de cada uma das linhas
182:     for(i=0;i<2;i++){
183:         if (simplex[i][colunaMenor] == 0){
184:             pp[i] = bigM;
185:         }else{
186:             pp[i] = simplex[i][6] / simplex[i][colunaMenor];
187:         }
188:     }
189:
190: //Dividir a linha por um valor específico obtendo o número 1
191:     if(pp[0] < pp[1]){
192:         strcpy(nomelinha[0], futuroNome);
193:         divisor0 = simplex[0][colunaMenor];
194:         for(i=0;i<7;i++){
195:             simplex[0][i] = simplex[0][i] / divisor0;
196:         }
197:         divisor1 = simplex[1][colunaMenor];
198:         for(i=0;i<7;i++){
199:             simplex[1][i] = simplex[1][i] - simplex[0][i]*divisor1;
200:         }
201:         divisor2 = simplex[2][colunaMenor];
202:         for(i=0;i<7;i++){
203:             simplex[2][i] = simplex[2][i] - simplex[0][i]*divisor2;
204:         }
205:     }else{

```

```

206:         strcpy(nomelinha[1], futuroNome);
207:         divisor0 = simplex[1][colunaMenor];
208:         for(i=0;i<7;i++){
209:             simplex[1][i] = simplex[1][i] / divisor0;
210:         }
211:         divisor1 = simplex[0][colunaMenor];
212:         for(i=0;i<7;i++){
213:             simplex[0][i] = simplex[0][i] - simplex[1][i]*divisor1;
214:         }
215:         divisor2 = simplex[2][colunaMenor];
216:         for(i=0;i<7;i++){
217:             simplex[2][i] = simplex[2][i] - simplex[1][i]*divisor2;
218:         }
219:     }
220:
221:     imprime(simplex, 3, 7, nomelinha, tipoMatriz);
222:     system ("pause");
223: }
224:
225:     printf("\nSolucoes obtidas atraves da analise do quadro otimo obtido : ");
226:     printf("\nA variavel %s assume o valor %.2f correspondente a %.f avioes do tipo",
A", nomelinha[0], simplex[0][6], ceil(simplex[0][6]));
227:     printf("\nA variavel %s assume o valor %.2f correspondente a %.f avioes do tipo",
B", nomelinha[1], simplex[1][6], ceil(simplex[1][6]));
228:     printf("\nO Valor final de Z no quadro otimo : %.2f", simplex[2][6]);
229:     printf("\nLogo o valor final do objetivo (Zmin) : %.2f", -1*simplex[2][6]);
230:     printf("\n\nSOLUCAO : Foi obtido um custo minimo de R$ %.2f ,\n", -
1*simplex[2][6]);
231:     printf("com o aviao A fazendo %.f viagens e o aviao B fazendo %.f",
viagens.\n\n\n\n\n", ceil(simplex[0][6]), ceil(simplex[1][6]));
232: }

```