

## Questão 02.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  #define LINHAS 4
6  #define COLUNAS 4
7
8  void geraMatriz(int matriz[LINHAS][COLUNAS]) {
9      srand(time(NULL));
10     for(int i = 0; i < LINHAS; i++) {
11         for(int j = 0; j < COLUNAS; j++) {
12             matriz[i][j] = rand() % 100;
13         }
14     }
15 }
16
17 void geraMatrizTransposta(int matriz[LINHAS][COLUNAS], int matrizTransposta[LINHAS][COLUNAS]) {
18     for(int i = 0; i < LINHAS; i++) {
19         for(int j = 0; j < COLUNAS; j++) {
20             matrizTransposta[j][i] = matriz[i][j];
21         }
22     }
23 }
24
25 void multiplicaMatriz(int matriz[LINHAS][COLUNAS], int matrizTransposta[LINHAS][COLUNAS], int
matrizResposta[LINHAS][COLUNAS]) {
26     for(int i = 0; i < LINHAS; i++) {
27         for(int j = 0; j < COLUNAS; j++) {
28
29             matrizResposta[i][j] = 0;
30
31             for (int k = 0; k < LINHAS; k++) {
32                 matrizResposta[i][j] += matriz[i][k] * matrizTransposta[k][j];
33             }
34
35         }
36     }
37 }
38
39 int verificaOrtogonal(int matrizResposta[LINHAS][COLUNAS]) {
40     for(int i = 0; i < LINHAS; i++) {
41         for(int j = 0; j < COLUNAS; j++) {
42             if(i==j) {
43                 if(matrizResposta[i][j] == 1)
44                     continue;
45                 else
46                     return 0;
47             } else {
48                 if(matrizResposta[i][j] == 0)
49                     continue;
50                 else
51                     return 0;
52             }
53
54         }
55     }
56     return 1;
57 }
```

```

58
59 int main() {
60
61     int matriz[LINHAS][COLUNAS];
62     int matrizTransposta[LINHAS][COLUNAS];
63     int matrizResposta[LINHAS][COLUNAS];
64
65     //Função 1 -> Criar Matriz:
66     geraMatriz(matriz);
67
68     //Função 2 -> Calcular Transposta:
69     geraMatrizTransposta(matriz, matrizTransposta);
70
71     //Função 3 -> Multiplicar Matrizes:
72     multiplicaMatriz(matriz, matrizTransposta, matrizResposta);
73
74     //Função 4 -> Verificar se a matriz é Ortogonal:
75     int resposta = 0;
76     resposta = verificaOrtogonal(matrizResposta);
77
78     //Exibindo Matriz:
79     printf("\n\n-----Matriz Gerada Aleatoriamente-----\n");
80     for(int i = 0; i < LINHAS; i++) {
81         for(int j = 0; j < COLUNAS; j++) {
82             printf(" %02d ", matriz[i][j]);
83         }
84         printf("\n");
85     }
86
87     //Exibindo Matriz Transposta:
88     printf("-----Matriz Transposta-----\n");
89     for(int i = 0; i < LINHAS; i++) {
90         for(int j = 0; j < COLUNAS; j++) {
91             printf(" %02d ", matrizTransposta[i][j]);
92         }
93         printf("\n");
94     }
95
96     //Exibindo Matriz Resposta:
97     printf("-----Matriz Resposta-----\n");
98     for(int i = 0; i < LINHAS; i++) {
99         for(int j = 0; j < COLUNAS; j++) {
100             printf(" %05d ", matrizResposta[i][j]);
101         }
102         printf("\n");
103     }
104
105     printf("-----\n");
106
107     if(resposta == 1)
108         printf("Matriz Ortogonal!");
109     else
110         printf("Matriz Nao Ortogonal!");
111
112     printf("\n-----\n\n");
113
114     return 0;
115 }

```