

```

1  /*
2  Implemente duas matrizes 10X10 cujos valores serão
3  informados pelo usuário em tempo de execução. Sobre
4  estas matrizes, faça:
5
6  (a) somar as duas matrizes e armazenar o resultado em
7  uma terceira matriz, exibindo os valores desta 3ª ma-
8  triz;
9
10 (b) verificar quantas vezes o último elemento da 1ª
11 matriz se repete na 2ª matriz;
12
13 (c) verificar se alguma das matrizes é nula ou se são
14 iguais.
15 */
16 #include <stdio.h>
17 #include <stdlib.h>
18 #define tam 2
19 int main(){
20     int A_matrix[tam][tam], *pmA = NULL;
21     int B_matrix[tam][tam], *pmB = NULL;
22     int C_matrix[tam][tam], *pmC = NULL;
23     int rptnumA1 = 0, *prnA = NULL;
24     int count_null_element_A_matrix = 0, *pcnemA = NULL;
25     int count_null_element_B_matrix = 0, *pcnemB = NULL;
26     int count_equal_element_AB_matrix = 0, *pceqAB = NULL;
27
28     pmA = &A_matrix[0][0];
29     pmB = &B_matrix[0][0];
30     pmC = &C_matrix[0][0];
31     prnA = &rptnumA1;
32     pcnemA = &count_null_element_A_matrix;
33     pcnemB = &count_null_element_B_matrix;
34     pceqAB = &count_equal_element_AB_matrix;
35
36     printf("\n\n\tEntre com os dados da matriz A:\n");
37     for(int i = 0; i < tam*tam; i++){
38         printf("\n\t\telemento de %d: ", i);
39         scanf("%d", pmA + i);
40     }
41
42     printf("\n\n\tEntre com os dados da matriz B:\n");
43     for(int i = 0; i < tam*tam; i++){
44         printf("\n\t\telemento de %d: ", i);
45         scanf("%d", pmB + i);
46     }
47     printf("\n\n\tOs elementos da matriz A sao:\n");

```

```

48     for(int i = 0; i < tam*tam; i++){
49         printf("\t\t%d\n", *(pmA + i) );
50     }
51     printf("\n\n\t0s elementos da matriz A sao:\n");
52     for(int i = 0; i < tam*tam; i++){
53         printf("\t\t%d\n", *(pmB + i) );
54     }
55
56     printf("\n\n\tA soma das matrizes A e B resulta na matriz C:\n"
57 );
58     for(int i = 0; i < tam*tam; i++){
59         *(pmC + i) = *(pmA + i) + *(pmB + i);
60     }
61     for(int i = 0; i < tam*tam; i++){
62         printf("\t\t%d\n", *(pmC + i) );
63     }
64
65     for(int i = 0; i < tam*tam; i++){
66         if( *(pmB + i) == *(pmA + tam*tam - 1)){
67             (*prnA)++;
68         }
69     }
70     printf("\n\n\t0 ultimo elemento da matriz A se repete %d vezes
71 na matriz B.\n", *prnA);
72
73     for(int i = 0; i < tam*tam; i++){
74         if( *(pmA + i) == 0 ){
75             (*pcnemA)++;
76         }
77     }
78     for(int i = 0; i < tam*tam; i++){
79         if( *(pmB + i) == 0 ){
80             (*pcnemB)++;
81         }
82     }
83     for(int i = 0; i < tam*tam; i++){
84         if( *(pmB + i) == *(pmA + i) ){
85             (*pceqAB)++;
86         }
87     }
88     if( (*pcnemA == tam*tam) ){
89         printf("\n\n\tA matriz 'A' eh nula.");
90     }
91     if( (*pcnemB == tam*tam) ){
92         printf("\n\n\tA matriz 'B' eh nula.");
93     }
94     if( (*pceqAB == tam*tam) ){

```

```
93         printf("\n\n\tAs matrizes 'A' e 'B' sao iguais.");
94     }
95     printf("\n");
96     return 0;
97 }
```