

N.º USP:

<input type="checkbox"/> 0								
<input type="checkbox"/> 1								
<input type="checkbox"/> 2								
<input type="checkbox"/> 3								
<input type="checkbox"/> 4								
<input type="checkbox"/> 5								
<input type="checkbox"/> 6								
<input type="checkbox"/> 7								
<input type="checkbox"/> 8								
<input type="checkbox"/> 9								

← Por favor codifique seu Número USP na esquerda e escreva seu nome abaixo.

Nome e sobrenome:

Question [q1] ♣ (2 pontos) Os trechos de código das Listagens 1, 2, 3, 4, 5, 6 e 7 tem sido proposto para para gerar a matriz *transposta* de uma *matriz* de dimensão $n \times m$, na qual n é sempre diferente de m ($n \neq m$). Marque (X) no número das Listagens que cumprem esse propósito.

Exemplo de `matriz[3][4]`: Conteúdo esperado para a matriz (`transposta[4][3]`):

0 1 2 3	0 4 8
4 5 6 7	1 5 9
8 9 0 1	2 6 0
	3 7 1

- 1 2 3 4 5 6 7
 Nenhuma das alternativas é correta

Question [q2] ♣ (2 pontos) Você foi escalado para desenvolver um trecho de programa para um caixa eletrônico. Este caixa, no entanto, fornece somente notas de 10, 20 e 50 reais. Considere que a quantidade solicitada pelo usuário está na variável "amount". Seu trecho de programa deve imprimir quantas notas de 10, 20 ou 50 devem ser liberadas de modo a minimizar o número de notas. Se não for possível liberar o valor pedido, uma mensagem indicando o fato deve ser dada ao usuário. Marque (X) na ordem adequada das linhas apresentadas na Listagem 8 para criar esse trecho de programa (as variáveis `quant` e `amount` são sempre inteiros positivos).

- Linhas: 10, 9, 11, 3, 8, 5, 2, 7, 4, 1, 6, 12
- Linhas: 10, 9, 11, 3, 8, 5, 1, 6, 2, 7, 4, 12
- Linhas: 10, 9, 11, 2, 7, 4, 3, 8, 5, 1, 6, 12
- Linhas: 10, 9, 11, 2, 7, 4, 1, 6, 3, 8, 5, 12
- Linhas: 10, 9, 11, 1, 6, 2, 7, 4, 3, 8, 5, 12
- Linhas: 10, 9, 11, 1, 6, 3, 8, 5, 2, 7, 4, 12
- Linhas: 10, 3, 8, 5, 2, 7, 4, 1, 6, 11, 9, 12
- Linhas: 10, 3, 8, 5, 1, 6, 2, 7, 4, 11, 9, 12
- Linhas: 10, 2, 7, 4, 3, 8, 5, 1, 6, 11, 9, 12
- Linhas: 10, 2, 7, 4, 1, 6, 3, 8, 5, 11, 9, 12
- Linhas: 10, 1, 6, 2, 7, 4, 3, 8, 5, 11, 9, 12
- Linhas: 10, 1, 6, 3, 8, 5, 2, 7, 4, 11, 9, 12
- Nenhuma das alternativas é correta

Question [q3] ♣ (1.5 pontos) O trecho de código da Listagem 9 tem sido proposto para processar um vetor chamado `nro_USP` que contém os dígitos de um número USP. Marque (X) nas afirmativas verdadeiras em relação ao conteúdo do vetor `nro_USP` depois que o código for executado.

- `nro_USP` é {9,7,6,3,2,2,1,0} se `tam=8` e o vetor inicial `nro_USP` é {1,0,2,7,6,9,3,2}
- `nro_USP` é {0,1,2,2,3,6,7,9} se `tam=8` e o vetor inicial `nro_USP` é {1,0,2,7,6,9,3,2}
- `nro_USP` é {3,9,6,7,2,2,0,1} se `tam=8` e o vetor inicial `nro_USP` é {1,0,2,7,6,9,3,2}
- `nro_USP` é {1,0,2,7,6,9,3,2} se `tam=8` e o vetor inicial `nro_USP` é {1,0,2,7,6,9,3,2}
- `nro_USP` é {0,1,2,5,6,6,7,7} se `tam=8` e o vetor inicial `nro_USP` é {1,0,2,7,6,6,7,5}
- `nro_USP` é {7,7,6,6,5,2,1,0} se `tam=8` e o vetor inicial `nro_USP` é {1,0,2,7,6,6,7,5}
- `nro_USP` é {5,6,6,7,7,2,0,1} se `tam=8` e o vetor inicial `nro_USP` é {1,0,2,7,6,6,7,5}
- `nro_USP` é {1,0,2,7,6,6,7,5} se `tam=8` e o vetor inicial `nro_USP` é {1,0,2,7,6,6,7,5}
- `nro_USP` é {0,1,3,4,5,8,9} se `tam=7` e o vetor inicial `nro_USP` é {9,8,0,5,3,4,1}
- `nro_USP` é {9,8,5,4,3,1,0} se `tam=7` e o vetor inicial `nro_USP` é {9,8,0,5,3,4,1}
- `nro_USP` é {1,4,3,5,0,8,9} se `tam=7` e o vetor inicial `nro_USP` é {9,8,0,5,3,4,1}
- `nro_USP` é {9,8,0,5,3,4,1} se `tam=7` e o vetor inicial `nro_USP` é {9,8,0,5,3,4,1}
- Nenhuma das alternativas é correta

Question [q4] ♣ (1 ponto) Supondo que exista um vetor de inteiros chamado `vet` de 50 posições (`int vet[50]`). Marque (X) nas opções que apresenta a descrição narrativa do trecho de código apresentado na Listagem 10.

- Os elementos da metade inferior do vetor são trocados de posição com os elementos da metade superior, mantendo sua ordem em cada metade, e todos os elementos são incrementados em uma unidade
- Os elementos da metade inferior do vetor são trocados de posição com os elementos da metade superior, mantendo sua ordem em cada metade
- Os elementos da metade inferior do vetor são trocados de posição com os elementos da metade superior, sem manter necessariamente a ordem em cada metade, e todos os elementos são incrementados em uma unidade
- O elemento da posição 0 é trocado com o elemento da posição 25 e é feito o incremento de uma unidade em ambos elementos, logo, o elemento da posição 1 é trocado com o elemento da posição 26 e é feito o incremento de uma unidade em ambos elementos, e assim por diante até o elemento da posição 24
- O elemento da posição 1 é trocado com o elemento da posição 25 e é feito o incremento de uma unidade em ambos elementos, logo, o elemento da posição 2 é trocado com o elemento da posição 26 e é feito o incremento de uma unidade em ambos elementos, e assim por diante até o elemento da posição 25
- O elemento da posição 0 é trocado com o elemento da posição 25, logo, o elemento da posição 1 é trocado com o elemento da posição 26, e assim por diante até o elemento da posição 24
- Nenhuma das alternativas é correta

CATALOG

Question [q5] ♣ (2 pontos) O programa apresentado na Listagem 11 tem sido proposto para identificar o valor de uma matriz $n \times m$ que mais se repete. O programa deve imprimir: o valor que teve o maior número de repetições e a quantidade de vezes que o elemento está na matriz.

Exemplo para a matriz [3] [4]:

```
0 1 2 3
4 0 1 2
3 4 0 1
```

Saída esperada do programa:

Valor 0 foi encontrado 3 vezes

Marque (X) nas modificações que, em conjunto, façam o programa funcionar adequadamente.

Grupo de Alternativas A

- A linha 23 deve ser mudada para: `for (j2 = 1 ; j2 <= m ; j2++) {`
- A linha 24 deve ser mudada para: `if (matriz[i1][j1] == matriz[i2-1][j2-1]){`

Grupo de Alternativas B

- A linha 29 deve ser mudada para: `if (count > max_count) {`
- A linha 31 deve ser mudada para: `val = matriz[i1][j1];`
- A linha 22 deve ser mudada para: `for (i2 = i1 ; i2 < n ; i2++) {`
- A linha 23 deve ser mudada para: `for (j2 = 0 ; j2 < m ; j2++) {`
- A linha 24 deve ser mudada para: `if (matriz[i1][j1] == matriz[i2][j2]){`
- A linha 22 deve ser mudada para: `for (i2 = i1+1 ; i2 < n ; i2++) {`
- A linha 23 deve ser mudada para: `for (j2 = j1+1 ; j2 < m ; j2++) {`
- A linha 24 deve ser mudada para: `if (matriz[i1][j1] == matriz[i2+1][j2+1]){`
- A linha 29 deve ser mudada para: `if (count < max_count) {`
- A linha 31 deve ser mudada para: `val = matriz[i1][j2];`
- A linha 24 deve ser mudada para: `val = matriz[i2][j2];`
- Nenhuma das alternativas é correta

Question [q6] (2 pontos) As olimpíadas terminaram, mas o trem-bala e seu sistema de gerenciamento ainda estão em projeto. Parte do problema é que o simulador para avaliar a movimentação de passageiros do trem-bala está incompleto. Você (aluno na melhor universidade do Brasil) foi designado para completar o simulador. Escreva na caixa de abaixo o código necessário que deve substituir a parte que diz *escreva seu código aqui* na Listagem 12 para completar o simulador. O simulador deve funcionar de acordo com a **Descrição 1**.

w p c

```
for (i = 0; i < n; i++) {
    total_emb += embarques[i];
    total_des += desembarques[i];
    circulacoes[i] = embarques[i] + desembarques[i];
    passageiros[i] = embarques[i] - desembarques[i];
    if (i > 0) passageiros[i] += passageiros[i-1];
}
diferenca = total_emb - total_des;

// outra possible solucao

for (i = 0; i < n; i++) {
    total_emb += embarques[i];
    total_des += desembarques[i];
    circulacoes[i] = embarques[i] + desembarques[i];
    passageiros[i] = total_emb - total_des;
}
diferenca = total_emb - total_des;
```

```
1 for (i = n-1; i >= 0; i--) {
2     for (j = m-1; j >= 0; j--) {
3         transposta[i][j] = matriz[j][i];
4     }
5 }
```

Listagem 1: Trecho de código na Linguagem C

```
1 for (i = m-1; i >= 0; i--) {
2     for (j = n-1; j >= 0; j--) {
3         transposta[j][i] = matriz[i][j];
4     }
5 }
```

Listagem 2: Trecho de código na Linguagem C

```
1 for (i = 0; i < m; i++) {
2     for (j = 0; j < n; j++) {
3         transposta[j][i] = matriz[i][j];
4     }
5 }
```

Listagem 3: Trecho de código na Linguagem C

```
1 for (i = 0; i < n; i++) {
2     for (j = 0; j < m; j++) {
3         transposta[i][j] = matriz[j][i];
4     }
5 }
```

Listagem 4: Trecho de código na Linguagem C

```
1 for (i = m-1; i >= 0; i--) {
2     for (j = n-1; j >= 0; j--) {
3         transposta[i][j] = matriz[j][i];
4     }
5 }
```

Listagem 5: Trecho de código na Linguagem C

```
1 for (i = 0; i < m; i++) {
2     for(j = n-1; j >= 0; j--) {
3         transposta[j][i] = matriz[i][j];
4     }
5 }
```

Listagem 6: Trecho de código na Linguagem C

```
1 for (i = 0; i < n; i++) {
2     for(j = 0; j < m; j++) {
3         transposta[j][i] = matriz[i][j];
4     }
5 }
```

Listagem 7: Trecho de código na Linguagem C

```
1 quant = amount / 10;
2 quant = amount / 20;
3 quant = amount / 50;
4 amount = amount % 20;
5 amount = amount % 50;
6 printf("%d notas de 10", quant);
7 printf("%d notas de 20", quant);
8 printf("%d notas de 50", quant);
9 printf("Não e possivel liberar o valor pedido!!!");
10 if (amount % 10 != 0) {
11 } else {
12 }
```

Listagem 8: Linhas de código na linguagem C a serem ordenadas

```
1 if (nro_USP[5] % 2 == 0) {
2     for (i=1; i < tam; i++) {
3         aux = nro_USP[i];
4         for (j = i-1; j >= 0 && nro_USP[j] > aux; j--)
5             nro_USP[j+1] = nro_USP[j];
6         nro_USP[j+1] = aux;
7     }
8 } else {
9     for (i=1; i < tam; i++) {
10        aux = nro_USP[i];
11        for (j = i-1; j >= 0 && nro_USP[j] < aux; j--)
12            nro_USP[j+1] = nro_USP[j];
13        nro_USP[j+1] = aux;
14    }
15 }
```

Listagem 9: Trecho de código para processar o vetor nro_USP

```
1 for (i = 0; i < 25; i++) {
2     temp = vet[i];
3     vet[i] = vet[i+25] + 1;
4     vet[i+25] = temp + 1;
5 }
```

Listagem 10: Trecho de código para processar o vetor vet

```

1  #include <stdio.h>
2
3  #define MAX 100
4
5  int main() {
6      int n, m, i, j;
7      int matriz[MAX][MAX];
8
9      printf("Ingrese o tamanho da matriz n x m: ");
10     scanf("%d %d", &n, &m);
11     for (i = 0 ; i < n ; i++) {
12         for (j = 0 ; j < m ; j++) {
13             printf("Valor da posição [%d][%d]: ", i, j);
14             scanf("%d", &matriz[i][j]);
15         }
16     }
17
18     int i1, j1, i2, j2, val, count, max_count = -1;
19     for (i1 = 0 ; i1 < n ; i1++) {
20         for (j1 = 0 ; j1 < m ; j1++) {
21             count = 0;
22             for (i2 = 1 ; i2 <= n ; i2++) {
23                 for (j2 = j1 ; j2 < m ; j2++) {
24                     if (matriz[i1][j1] != matriz[i2][j2]) {
25                         count++;
26                     }
27                 }
28             }
29             if (count <= max_count) {
30                 max_count = count;
31                 val = matriz[i2][j1];
32             }
33         }
34     }
35     printf("Valor %d foi encontrado %d vezes", val, max_count);
36
37     return 0;
38 }

```

Listagem 11: Código do programa para calcular e contar o valor mais repetido em uma matriz $n \times m$ de tamanho máximo 100 x 100

```

1  #include <stdio.h>
2
3  #define MAX 20
4
5  int main() {
6      char estacoes[MAX][20]; // vetor com os nomes das estações
7      int embarques[MAX];
8      int desembarques[MAX];
9      int circulacoes[MAX];
10     int passageiros[MAX];
11
12     int n, i, j, total_emb = 0, total_des = 0, diferenca = 0;
13
14     printf("Numero de estações? ");
15     scanf("%d", &n);
16     for (i = 0 ; i < n ; i++) {
17         printf("Nome da estação %d? ", i);
18         scanf("%s", estacoes[i]);
19         printf("Quantos passageiros embarcaram? ");
20         scanf("%d", &embarques[i]);
21         printf("Quantos passageiros desembarcaram? ");
22         scanf("%d", &desembarques[i]);
23     }
24
25     // ... escreva seu código aqui
26
27     printf("Total de embarques: %d\n", total_emb);
28     printf("Total de desembarques: %d\n", total_des);
29     printf("Circulação de passageiros por estação:\n");
30     for (i = 0; i < n; i++) {
31         printf(" Estação %d (%s): %d\n", i, estacoes[i],
32             circulacoes[i]);
33     }
34     printf("Passageiros transportados por estação:\n");
35     for (i = 0; i < n; i++) {
36         printf(" Estação %d (%s): %d\n", i, estacoes[i],
37             passageiros[i]);
38     }
39     printf("Diferença entre Embarque-Desembarques: %d\n",
40         diferenca);
41
42     return 0;
43 }

```

Listagem 12: Código incompleto de programa na linguagem C para o simulador de movimentação de passageiros do trem-bala

Descrição 1. O simulador deve coletar as informações de quantas pessoas embarcaram e desembarcaram em cada uma das estações (controlados pelas catracas eletrônicas), realizando uma estatística final da movimentação de passageiros: total de passageiros embarcados, total de passageiros desembarcados, circulação de passageiros por estação (pessoas que embarcaram + pessoas que desembarcaram), passageiros transportados por estação (dentro do trem ao sair da estação) e a diferença total entre os embarques e desembarques (se a diferença não é 0, há problema no controle de passageiros). Os dados de movimentação (embarques e desembarque) são armazenados em vetores. A seguir, apresentamos dois exemplo no simulador para o trecho Campinas - Rio de Janeiro com estações em: Campinas [Estação 0], SãoPaulo [Estação 1] SãoJose [Estação 2], Resende [Estação 3] e Rio [Estação 4].

Dados de entrada para o simulador:

Numero de estações? 5
 Nome da estação 0? Campinas
 Quantos passageiros embarcaram? 1234
 Quantos passageiros desembarcaram? 100
 Nome da estação 1? SãoPaulo
 Quantos passageiros embarcaram? 1234
 Quantos passageiros desembarcaram? 1000
 Nome da estação 2? SãoJose
 Quantos passageiros embarcaram? 1234
 Quantos passageiros desembarcaram? 500
 Nome da estação 3? Resende
 Quantos passageiros embarcaram? 500
 Quantos passageiros desembarcaram? 2250
 Nome da estação 4? Rio
 Quantos passageiros embarcaram? 0
 Quantos passageiros desembarcaram? 300

Numero de estações? 5
 Nome da estação 0? Campinas
 Quantos passageiros embarcaram? 1234
 Quantos passageiros desembarcaram? 100
 Nome da estação 1? SãoPaulo
 Quantos passageiros embarcaram? 1234
 Quantos passageiros desembarcaram? 1000
 Nome da estação 2? SãoJose
 Quantos passageiros embarcaram? 1234
 Quantos passageiros desembarcaram? 800
 Nome da estação 3? Resende
 Quantos passageiros embarcaram? 500
 Quantos passageiros desembarcaram? 1550
 Nome da estação 4? Rio
 Quantos passageiros embarcaram? 0
 Quantos passageiros desembarcaram? 752

Saída esperada do simulador:

Total de embarques: 4202
 Total de desembarques: 4150
 Circulação de passageiros por estação:
 Estação 0 (Campinas): 1334
 Estação 1 (SãoPaulo): 2234
 Estação 2 (SãoJose): 1734
 Estação 3 (Resende): 2750
 Estação 4 (Rio): 300
 Passageiros transportados por estação:
 Estação 0 (Campinas): 1134
 Estação 1 (SãoPaulo): 1368
 Estação 2 (SãoJose): 2102
 Estação 3 (Resende): 352
 Estação 4 (Rio): 52
 Diferença entre Embarque-Desembarques: 52

Total de embarques: 4202
 Total de desembarques: 4202
 Circulação de passageiros por estação:
 Estação 0 (Campinas): 1334
 Estação 1 (SãoPaulo): 2234
 Estação 2 (SãoJose): 2034
 Estação 3 (Resende): 2050
 Estação 4 (Rio): 752
 Passageiros transportados por estação:
 Estação 0 (Campinas): 1134
 Estação 1 (SãoPaulo): 1368
 Estação 2 (SãoJose): 1802
 Estação 3 (Resende): 752
 Estação 4 (Rio): 0
 Diferença entre Embarque-Desembarques: 0