

5.2 Exercícios

Exercício 5.1: Escreva um programa que preencha um array de dimensões 3x2 de inteiros com valores fornecidos pelo usuário e o exiba na forma de uma matriz.

Arquivo com a solução: [ex5.1.c](#)

Entrada

```
array[0][0] : 1  
array[0][1] : 2  
array[1][0] : 3  
array[1][1] : 4  
array[2][0] : 5  
array[2][1] : 6
```

Saída

```
001 002  
003 004  
005 006
```

Exercício 5.2: Escreva um programa que preencha dois arrays de dimensões 3x3 de inteiros com valores fornecidos pelo usuário e armazene a soma desses dois arrays em um terceiro array de dimensões 3x3. No final, o programa deve exibir os três arrays no formato apresentado a seguir.

Arquivo com a solução: [ex5.2.c](#)

Entrada

```
array1[0][0]: 4
array1[0][1]: 7
array1[0][2]: 8
array1[1][0]: 5
array1[1][1]: 1
array1[1][2]: 2
array1[2][0]: 6
array1[2][1]: 5
array1[2][2]: 8
array2[0][0]: 9
array2[0][1]: 5
array2[0][2]: 2
array2[1][0]: 1
array2[1][1]: 4
array2[1][2]: 5
array2[2][0]: 6
array2[2][1]: 3
array2[2][2]: 2
```

Saída

```
array1:*****array2:*****arraySoma:
004 007 008***009 005 002***013 012 010
005 001 002 + 001 004 005 = 006 005 007
006 005 008***006 003 002***012 008 010
```

Exercício 5.3: Escreva um programa que preencha um array de dimensões 3x4 de inteiros com valores fornecidos pelo usuário. Em seguida, o programa deve ler o valor de um número inteiro. Armazene em um segundo array de dimensões 3x4 a multiplicação do valor fornecido pelas posições do array preenchido inicialmente. No final, o programa deve exibir o array contendo a multiplicação na forma de uma matriz.

Arquivo com a solução: [ex5.3.c](#)

Entrada

```
array[0][0]: 1
array[0][1]: 4
array[0][2]: 5
array[0][3]: 8
array[1][0]: 7
array[1][1]: 4
array[1][2]: 5
array[1][3]: 2
array[2][0]: 3
array[2][1]: 6
array[2][2]: 5
array[2][3]: 4
Multiplicar por: 5
```

Saída

```
arrayMult:
005 020 025 040
035 020 025 010
015 030 025 020
```

Exercício 5.4:

Escreva um programa que preencha um array de dimensões 2x2 de inteiros com valores fornecidos pelo usuário. O programa deve calcular e exibir o determinante da matriz representada por esse array. Lembrando que:

- Para $M_{2 \times 2} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$,
- $D = a_{1,1} \cdot a_{2,2} - (a_{1,2} \cdot a_{2,1})$
- Onde:
 - $M_{2 \times 2}$ é uma matriz de dimensões 2x2;
 - D é o determinante dessa matriz;
 - **Obs:** cuidado com os índices do array, pois iniciam em 0 e não 1.

Arquivo com a solução: [ex5.4.c](#)

Entrada

```
array[0][0] : 4
array[0][1] : 5
array[1][0] : 6
array[1][1] : 1
```

Saída

Determinante: -26

Exercício 5.5:

Escreva um programa que preencha um array de dimensões 3x3 de inteiros com valores fornecidos pelo usuário. O programa deve calcular e exibir o determinante da matriz representada por esse array. Lembrando que:

- Para $M_{3 \times 3} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$,
- $D = a_{1,1} \cdot a_{2,2} \cdot a_{3,3} + a_{1,2} \cdot a_{2,3} \cdot a_{3,1} + a_{1,3} \cdot a_{2,1} \cdot a_{3,2} - (a_{1,3} \cdot a_{2,2} \cdot a_{3,1} + a_{1,1} \cdot a_{2,3} \cdot a_{3,2} + a_{1,2} \cdot a_{2,1} \cdot a_{3,3})$
- Onde:
 - $M_{3 \times 3}$ é uma matriz de dimensões 3x3;
 - D é o determinante dessa matriz;
 - **Obs:** cuidado com os índices do array, pois iniciam em 0 e não 1.

Arquivo com a solução: [ex5.5.c](#)**Entrada**

```
array[0][0] : 4
array[0][1] : 5
array[0][2] : 7
array[1][0] : 8
array[1][1] : 2
array[1][2] : 1
array[2][0] : 3
array[2][1] : 6
array[2][2] : 5
```

Saída

Determinante: 125

Exercício 5.6:

Escreva um programa que preencha um array de dimensões 2x3 de inteiros com valores fornecidos pelo usuário. Esse array será considerado como uma matriz. O programa deve preencher um segundo array de dimensões 3x2 com os valores que representem a matriz transposta da matriz contida do primeiro array. Por fim, o programa deve exibir a matriz original e a matriz transposta. Lembrando que:

- Para $M_{2 \times 3} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix}$,
- $M^t = \begin{bmatrix} a_{1,1} & a_{2,1} \\ a_{1,2} & a_{2,2} \\ a_{1,3} & a_{2,3} \end{bmatrix}$
- Onde:
 - $M_{2 \times 3}$ é uma matriz de dimensões 2x3;
 - M^t é a matriz transposta de M de dimensões 3x2;
 - **Obs:** cuidado com os índices do array, pois iniciam em 0 e não 1.

Arquivo com a solução: [ex5.6.c](#)

Entrada

```
array[0][0]: 1
array[0][1]: 2
array[0][2]: 3
array[1][0]: 4
array[1][1]: 5
array[1][2]: 6
```

Saída

```
M:
001 002 003
004 005 006

Mt:
001 004
002 005
003 006
```

Exercício 5.7:

Escreva um programa que preencha dois arrays de inteiros, um de dimensões 3x2 enquanto o outro de dimensões 2x3. As duas matrizes representadas pelos arrays devem ser multiplicadas e o resultado deve ser armazenado em um terceiro array bidimensional de dimensões 3x3. Por fim, o programa deve exibir o array que contém a multiplicação. Lembrando que:

- Para $A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{bmatrix}$ e
- $B = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \end{bmatrix}$,
- $A \cdot B = \begin{bmatrix} a_{1,1} \cdot b_{1,1} + a_{1,2} \cdot b_{2,1} & a_{1,1} \cdot b_{1,2} + a_{1,2} \cdot b_{2,2} & a_{1,1} \cdot b_{1,3} + a_{1,2} \cdot b_{2,3} \\ a_{2,1} \cdot b_{1,1} + a_{2,2} \cdot b_{2,1} & a_{2,1} \cdot b_{1,2} + a_{2,2} \cdot b_{2,2} & a_{2,1} \cdot b_{1,3} + a_{2,2} \cdot b_{2,3} \\ a_{3,1} \cdot b_{1,1} + a_{3,2} \cdot b_{2,1} & a_{3,1} \cdot b_{1,2} + a_{3,2} \cdot b_{2,2} & a_{3,1} \cdot b_{1,3} + a_{3,2} \cdot b_{2,3} \end{bmatrix}$
- Onde:
 - A é uma matriz de dimensões 3x2;
 - B é uma matriz de dimensões 2x3;
 - **Obs:** cuidado com os índices do array, pois iniciam em 0 e não 1.

Arquivo com a solução: [ex5.7.c](#)**Entrada**

```
array1[0][0]: 2
array1[0][1]: 3
array1[1][0]: 0
array1[1][1]: 1
array1[2][0]: -1
array1[2][1]: 4
array2[0][0]: 1
array2[0][1]: 2
array2[0][2]: 3
array2[1][0]: -2
array2[1][1]: 0
array2[1][2]: 4
```

Saída

```
A x B =  
-04 004 018  
-02 000 004  
-09 -02 013
```

Exercício 5.8 (URI..., 2020): Escreva um programa que leia um inteiro no intervalo de 1, inclusive, a 100, inclusive. Caso o valor fornecido esteja fora desse intervalo, o programa deve avisar o usuário e terminar. Caso contrário, esse valor deve ser usado para gerar uma saída de acordo com o padrão apresentado em cada exemplo. Os valores inteiros devem ser formatados alinhados à direita, com largura de três caracteres, preenchidos com espaços, usando para isso o especificador de formato %3d e separados por um espaço. Os asteriscos quase pretos indicam espaços. **Obs.:** Este exercício é uma adaptação do problema número 1435 do URI *Online Judge*.

Arquivo com a solução: [ex5.8.c](#)

Entrada

```
Numero entre 1 e 100: 1
```

Saída

```
**1
```

Entrada

```
Numero entre 1 e 100: 2
```

Saída

```
**1***1  
**1***1
```

Entrada

```
Numero entre 1 e 100: 3
```

Saída

```
**1***1***1
**1***2***1
**1***1***1
```

Entrada

Numero entre 1 e 100: 4

Saída

```
**1***1***1***1
**1***2***2***1
**1***2***2***1
**1***1***1***1
```

Entrada

Numero entre 1 e 100: 5

Saída

```
**1***1***1***1***1
**1***2***2***2***1
**1***2***3***2***1
**1***2***2***2***1
**1***1***1***1***1
```

Entrada

Numero entre 1 e 100: 10

Saída

```

**1**1**1**1**1**1**1**1**1**1**1**1
**1**2**2**2**2**2**2**2**2**2**2**1
**1**2**3**3**3**3**3**3**3**2**1
**1**2**3**4**4**4**4**4**3**2**1
**1**2**3**4**5**5**4**3**2**1
**1**2**3**4**5**5**4**3**2**1
**1**2**3**4**4**4**4**3**2**1
**1**2**3**3**3**3**3**3**2**1
**1**2**2**2**2**2**2**2**2**2**1
**1**1**1**1**1**1**1**1**1**1**1

```

Entrada

```
Numero entre 1 e 100: 0
```

Saída

```
Numero incorreto!
```

Entrada

```
Numero entre 1 e 100: 200
```

Saída

```
Numero incorreto!
```

Exercício 5.9 (URI..., 2020): Escreva um programa que leia um inteiro no intervalo de 1, inclusive, a 100, inclusive. Caso o valor fornecido esteja fora desse intervalo, o programa deve avisar o usuário e terminar. Caso contrário, esse valor deve ser usado para gerar uma saída de acordo com o padrão apresentado em cada exemplo. Os valores inteiros devem ser formatados alinhados à direita, com largura de três caracteres, preenchidos com espaços, usando para isso o especificador de formato `%3d` e separados por um espaço. Os asteriscos quase pretos indicam espaços. **Obs.:** Este exercício é uma adaptação do problema número 1478 do URI *Online Judge*.

Arquivo com a solução: [ex5.9.c](#)

Entrada

Numero entre 1 e 100: 1

Saída

**1

Entrada

Numero entre 1 e 100: 2

Saída

1*2

2*1

Entrada

Numero entre 1 e 100: 3

Saída

1*2***3

2*1***2

3*2***1

Entrada

Numero entre 1 e 100: 4

Saída

1*2***3***4

2*1***2***3

3*2***1***2

4*3***2***1

Entrada

Numero entre 1 e 100: 5

Saída

```
**1**2**3**4**5
**2**1**2**3**4
**3**2**1**2**3
**4**3**2**1**2
**5**4**3**2**1
```

Entrada

Numero entre 1 e 100: 10

Saída

```
**1**2**3**4**5**6**7**8**9**10
**2**1**2**3**4**5**6**7**8**9
**3**2**1**2**3**4**5**6**7**8
**4**3**2**1**2**3**4**5**6**7
**5**4**3**2**1**2**3**4**5**6
**6**5**4**3**2**1**2**3**4**5
**7**6**5**4**3**2**1**2**3**4
**8**7**6**5**4**3**2**1**2**3
**9**8**7**6**5**4**3**2**1**2
**10**9**8**7**6**5**4**3**2**1
```

Entrada

Numero entre 1 e 100: 0

Saída

Numero incorreto!

Entrada

Numero entre 1 e 100: 200

Saída

Numero incorreto!