
Professor: Camillo Falcão

camillofalcao@gmail.com

Exercícios de Fixação

Algoritmos: vetores, strings e matrizes.

OBJETIVO

O objetivo desta lista de exercícios é auxiliar o aluno no entendimento dos principais conceitos relacionados a vetores, strings e matrizes.

DICAS

1. Tente fazer os exercícios simulado sozinho(a) e sem consulta.
2. Se você não conseguir terminar todas as questões, utilize o material da disciplina como fonte de consulta, mas lembre-se de que a nossa prova será sem consulta.

Questão 1:

Assinale a alternativa que contém o código que imprime o vetor (v) em tela.

a) `Console.WriteLine("{0} ", v);`

b) `for (int i = 0; i < v.Length; i++)
 v[i] = Convert.ToInt32(Console.ReadLine());`

c) `for (int i = 0; i > v.Length; i++)
 Console.Write("{0} ", v[i]);`

☒ d) `for (int i = 0; i < v.Length; i++)
 Console.Write("{0} ", v[i]);`

Questão 2:

Assinale a alternativa que contém o código que lê o vetor (v) de números inteiros.

- a) `Console.WriteLine("{0} ", v);`
- ☒ b) `for (int i = 0; i < v.Length; i++)
v[i] = Convert.ToInt32(Console.ReadLine());`
- c) `for (int i = 0; i > v.Length; i++)
Console.Write("{0} ", v[i]);`
- d) `for (int i = 0; i < v.Length; i++)
Console.Write("{0} ", v[i]);`

Questão 3:

Veja o trecho de código abaixo e assinale a alternativa que indica quais valores estarão armazenados no vetor3.

```
int[] vetor1 = new int[] { 1, 2, 3 };  
int[] vetor2 = new int[] { 2, 2, 4 };  
int[] vetor3 = new int[3];  
for (int i = 0; i < vetor3.Length; i++)  
    vetor3[i] = vetor1[i] + vetor2[i];
```

- a) { 1, 2, 3 }
- b) { 2, 2, 4 }
- ☒ c) { 3, 4, 7 }
- d) { 0, 0, 0 }

Questão 4:

Veja o trecho de código abaixo e assinale a alternativa que indica quais valores estarão armazenados no vetor3.

```
int i;  
int[] vetor1 = new int[] { 1, 2, 3 };  
int[] vetor2 = new int[] { 2, 2, 4 };  
int[] vetor3 = new int[6];  
for (i = 0; i < vetor1.Length; i++)  
    vetor3[i] = vetor1[i];  
for (i = 0; i < vetor2.Length; i++)  
    vetor3[vetor2.Length + i] = vetor2[i];
```

- a) { 1, 2, 3 }
- ☒ b) { 1, 2, 3, 2, 2, 4 }
- c) { 2, 2, 4, 1, 2, 3 }
- d) { 0, 0, 0, 0, 0, 0 }

Questão 5:

Veja o trecho de código abaixo e assinale a alternativa que indica qual a finalidade do mesmo.

```
static void X(double[] v)
{
    for (int i = 0; i < v.Length; i++)
    {
        v[i] = Convert.ToDouble(Console.ReadLine());
        if (v[i] < 0)
            v[i] = v[i] * -1;
    }
}
```

- a) Multiplica todos os elementos de um vetor por -1.
- b) Lê uma string informada pelo usuário e inverte as posições dos seus elementos, mas apenas dos elementos que possuem representação gráfica.
- ☒ c) Lê um vetor de números reais e retira o sinal de negativo dos valores menores que zero informados pelo usuário.
- d) Imprime os elementos de um vetor de números reais passados por parâmetro.

Questão 6:

Veja o trecho de código abaixo e assinale a alternativa que indica o que será impresso ao término da sua execução.

```
string texto = "Teste isso";  
for (int i = texto.Length - 1; i >= 0; i--)  
    Console.Write(texto[i]);
```

- ☒ a) ossi etseT
- ☐ b) Teste isso
- ☐ c) T
- ☐ d) o

Questão 7:

Qual valor retornado ao chamar a função abaixo passando por parâmetro a string "Teste isso"?

```
static string Funcao(string texto)  
{  
    var sb = new StringBuilder();  
  
    for (int i = 0; i < texto.Length; i++)  
    {  
        if (texto[i] != ' ')  
            sb.Append(texto[i]);  
    }  
  
    return sb.ToString();  
}
```

- ☐ a) ossietseT
- ☒ b) Testeisso
- ☐ c) Teste isso
- ☐ d) ossi etseT

Questão 8:

A função abaixo possui uma estrutura para tratamento de erros e você não está familiarizado à mesma. Desconsidere o código que você não entender corretamente e responda: qual a finalidade do método de nome "Funcao" abaixo?

Observação: após responder a esta questão, experimente alterar a matriz para uma matriz não quadrada e veja o que ocorre.

```
try
{
    Console.WriteLine(Funcao(mat));
}
catch (Exception ex)
{
    Console.WriteLine("Erro: {0}", ex.Message);
}

static double Funcao(double[,] mat)
{
    if (mat.GetLength(0) != mat.GetLength(1))
    {
        throw new Exception("Não é possível <FINALIDADE DA "
            + "FUNÇÃO AQUI> em uma matriz não quadrada.");
    }

    double soma = 0;

    for (int i = 0; i < mat.GetLength(0); i++)
        soma += mat[i, i];

    return soma;
}
```

- a) A função retorna a soma dos elementos da matriz.
- b) A função retorna todos os elementos da matriz.
- c) A função retorna o primeiro elemento da matriz.
- d) A função retorna a soma dos elementos pertencentes à diagonal principal da matriz.**

Questão 9: Veja o código abaixo e assinale a alternativa que contém um comando de repetição que pode substituir o `for` do código abaixo sem alteração de semântica.

```
double[,] mat = new double[3,3] { { 1, 2, 3 },  
                                   { 4, 5, 6 },  
                                   { 6, 7, 8 } };
```

```
for (int i = 0, j=mat.GetLength(1)-1; i < mat.GetLength(0); i++, j--)  
    Console.WriteLine(mat[i, j]);
```

a)

```
for (int i = 0; i < mat.GetLength(0); i++)  
    soma += mat[i, i];
```

b)

```
for (int j = 0; j < mat.GetLength(0); j++)  
    soma += mat[j, j];
```

c)

```
for (int i = 0; i < mat.GetLength(0); i++)  
    Console.WriteLine(mat[i, mat.GetLength(1) - 1 - i]);
```

d)

```
for (int i = 0, j=mat.GetLength(1)-1; i < mat.GetLength(0); i++, j--)  
    Console.WriteLine(mat[j, i]);
```

Gabarito:

Questão 1: D

Questão 2: B

Questão 3: C

Questão 4: B

Questão 5: C

Questão 6: A

Questão 7: B

Questão 8: D

Questão 9: C