

Exercícios de Fixação - Lógica de Programação

Bóson Treinamentos em Tecnologia

Vetores e Matrizes

Crie algoritmos para os itens a seguir, testando os programas no Portugol Studio.

Exercício 1

Crie um algoritmo que leia 10 números informados pelo usuário, e os mostre na tela na ordem inversa da leitura, ou seja, o último número lido deve ser o primeiro mostrado.

Exercício 2

Crie um vetor de 5 números informados pelo usuário e monte outro vetor contendo os valores do primeiro multiplicados por um fator também informado pelo usuário. Mostrar os elementos de ambos os vetores na tela, lado a lado, como segue.

Usuário escolheu fator 3 (multiplicar por 3)

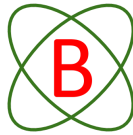
Número	x3
9	27
7	21
8	24
5	15
2	6

Dica: Use caractere de tabulação \t para deixar os números alinhados nas colunas

Exercício 3

Cadastrar 5 nomes de professores e seus respectivos tempos de profissão. Armazene esses valores em dois vetores distintos.

Mostrar os dados na tela, em uma tabela com os nomes dos professores na coluna à esquerda, e os tempos de profissão de cada um em uma coluna à direita, como no exemplo a seguir:



```
Digite o nome do professor: Fábio
Digite o tempo de casa do professor: 16
Digite o nome do professor: Ana
Digite o tempo de casa do professor: 20
Digite o nome do professor: Rita
Digite o tempo de casa do professor: 22
Digite o nome do professor: Jorge
Digite o tempo de casa do professor: 4
Digite o nome do professor: Mônica
Digite o tempo de casa do professor: 12
```

Professor	Tempo de Casa
Fábio	16
Ana	20
Rita	22
Jorge	4
Mônica	12

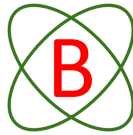
Exercício 4

Um depósito armazena 8 produtos distintos, e para cada tipo de produto existe um código. Crie um algoritmo para ler o código do produto e a quantidade em estoque.

Monte dois vetores para armazenar respectivamente os códigos das mercadorias e a quantidade dos produtos.

Mostrar na tela um relatório dos dados, em formato tabular, como no exemplo seguinte:

CodProduto	QtdeEstoque
TV-01	9
AS-03	7
RF-14	3
.	.
.	.



Exercício 5

Cadastrar 10 números de matrículas de alunos e armazená-los em um vetor. Os números devem estar no intervalo de 100 a 200, e não podem ser repetidos.

Se forem digitados números fora desse intervalo, exibir uma mensagem de erro e pedir para o usuário digitar novamente.

Mostrar na tela as matrículas armazenadas, em ordem crescente.

Exercício 6

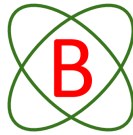
Gerar aleatoriamente números para preencher uma matriz 4x4, temporizando a saída de cada número, de modo que a cada 1 segundo um dos números gerados seja exibido na tela.

Dica: Para temporizar a execução de instruções, use a função *aguarde* da biblioteca Util.

Exercício 7

Elabore um algoritmo que preencha duas matrizes 4x4 lendo valores digitados pelo usuário, e então mostre na tela as matrizes completas, assim como a soma de ambas.

Dica: A soma de matrizes de mesmo tamanho é feita somando-se os valores de cada posição em uma matriz com os valores das posições correspondentes na outra matriz.



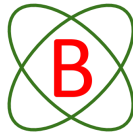
Resolução dos Exercícios

Exercício 1

```
programa
{
    inteiro vetor[10]

    funcao inicio()
    {
        para(inteiro i = 0; i < 10; i++) {
            escreva("Digite um número: ")
            leia(vetor[i])
        }

        para(inteiro i = 9; i >= 0; i--) {
            escreva(vetor[i] + " ")
        }
    }
}
```



Exercício 2

```
programa
{

inteiro vetor[5]
    inteiro vetorX3[5]
    inteiro fator

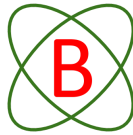
funcao inicio()
{

    // Preencher vetor 1
    para(inteiro i = 0; i < 5; i++) {
        escreva("Digite um número: ")
        leia(vetor[i])
    }

    escreva("Por quanto deseja multiplicar os valores? ")
    leia(fator)
    limpa()

    // Preencher vetor 2
    para(inteiro i = 0; i < 5; i++) {
        vetorX3[i] = vetor[i] * fator
    }

    // Mostrar resultados na tela
    escreva("Número\tx" + fator + "\n")
    para(inteiro i = 0; i < 5; i++) {
        escreva(vetor[i] + "\t" + vetorX3[i] + "\n")
    }
}
}
```



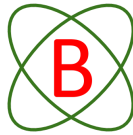
Exercício 3

```
programa
{
    cadeia nome[5]
    inteiro tempo[5]

    funcao inicio()
    {

        // Preencher vetor 1
        para(inteiro i = 0; i < 5; i++) {
            escreva("Digite o nome do professor: ")
            leia(nome[i])
            escreva("Digite o tempo de casa do professor: ")
            leia(tempo[i])
        }

        // Mostrar resultados na tela
        escreva("\nProfessor\tTempo de Casa\n")
        para(inteiro i = 0; i < 5; i++) {
            escreva(nome[i] + "\t\t" + tempo[i] + "\n")
        }
    }
}
```



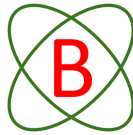
Exercício 4

```
programa
{
    cadeia codProd[8]
    inteiro qtde[8]

    funcao inicio()
    {

        // Preencher vetor 1
        para(inteiro i = 0; i < 8; i++) {
            escreva("Digite o código do produto: ")
            leia(codProd[i])
            escreva("Digite a quantidade em estoque: ")
            leia(qtde[i])
        }

        // Mostrar relatório
        escreva("CodProduto\tQtdeEstoque\n")
        para(inteiro i = 0; i < 8; i++) {
            escreva(codProd[i] + "\t\t" + qtde[i] + "\n")
        }
    }
}
```



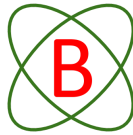
Exercício 5

```
programa {

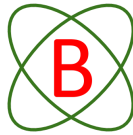
    inteiro alunos[10]
    inteiro pos, mat, cont, contA, contB, aux
    caracter controle = 's'

    funcao inicio() {
        para (pos = 0; pos < 10; pos++) {
            alunos[pos] = 0
        }
        para (pos = 0; pos < 10; pos++) {
            enquanto(verdadeiro) {
                escreva ("Digite o código de matrícula: ")
                leia(mat)
                para (cont = 0; cont < 10; cont++) {
                    se (mat == alunos[cont]) {
                        escreva("Valor já existe\n")
                        controle = 'n'
                    }
                }
                se ((mat < 100) ou (mat > 200)) {
                    escreva("Valor inválido\n")
                    controle = 'n'
                }
                se (controle != 'n') {
                    alunos[pos] = mat
                    pare
                }
                controle = 's'
            }
        }

        // Ordenar o array
        para (contA = 0; contA < 10; contA++) {
            para (contB = contA + 1; contB < 10; contB++) {
                se (alunos[contA] > alunos[contB]) {
                    aux = alunos[contB]
                    alunos[contB] = alunos[contA]
                    alunos[contA] = aux
                }
            }
        }
    }
}
```

```
        }  
    }  
}  
  
// Mostrar as matrículas ordenadas  
limpa()  
escreva("Matrículas realizadas: \n")  
para (contA = 0; contA < 10; contA++) {  
    escreva(alunos[contA] + " ")  
}  
  
}  
}
```



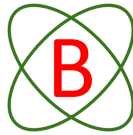
Exercício 6

```
programa
{
    inclua biblioteca Util --> u

    inteiro matriz[4][4] // Matriz quadrada 4x4
    inteiro lin, col

    funcao inicio()
    {
        para(lin = 0; lin <= 3; lin++) {
            para(col = 0; col <= 3; col++) {
                matriz[lin][col] = u.sorteia(0,9)
            }
        }

        // Escrever os dados da matriz na tela:
        escreva("\n")
        para(lin = 0; lin <= 3; lin++) {
            para(col = 0; col <= 3; col++) {
                se(col < 3) {
                    escreva(matriz[lin][col] + " ")
                    u.aguarde(1000)
                }
                senao {
                    escreva(matriz[lin][col] + "\n")
                    u.aguarde(1000)
                }
            }
        }
    }
}
```



Exercício 7

```
programa {
    // Criar três matrizes 4x4
    inteiro A[4][4]
    inteiro B[4][4]
    inteiro S[4][4]
    inteiro linha, coluna

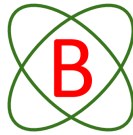
    funcao inicio() {
        escreva("Algoritmo para soma de matrizes 4x4\n")

        // Preencher as duas matrizes
        escreva("Preencher a matriz A:\n")
        para (linha = 0; linha <= 3; linha++) {
            para (coluna = 0; coluna <= 3; coluna++) {
                escreva("Digite um número para inserir na matriz A:\n")
                leia(A[linha][coluna])
            }
        }

        escreva("\nPreencher a matriz B:\n")
        para (linha = 0; linha <= 3; linha++) {
            para (coluna = 0; coluna <= 3; coluna++) {
                escreva("Digite um número para inserir na matriz B:\n")
                leia(B[linha][coluna])
            }
        }

        // Efetuar a soma das matrizes e armazenar o resultado na matriz S:
        para (linha = 0; linha <= 3; linha++) {
            para (coluna = 0; coluna <= 3; coluna++) {
                S[linha][coluna] = A[linha][coluna] + B[linha][coluna]
            }
        }

        // Exibir os resultados na tela:
        escreva("\nConteúdo da matriz A:\n")
        para (linha = 0; linha <= 3; linha++) {
            para (coluna = 0; coluna <= 3; coluna++) {
```



```
        se (coluna < 3) {
            escreva(A[linha][coluna] + " ")
        }
        senao {
            escreva(A[linha][coluna] + "\n")
        }
    }
}
```

```
escreva("\nConteúdo da matriz B:\n")
para (linha = 0; linha <= 3; linha++) {
    para (coluna = 0; coluna <= 3; coluna++) {
        se (coluna < 3) {
            escreva(B[linha][coluna] + " ")
        }
        senao {
            escreva(B[linha][coluna] + "\n")
        }
    }
}
```

```
escreva("\nA soma das matrizes A e B é:\n")
para (linha = 0; linha <= 3; linha++) {
    para (coluna = 0; coluna <= 3; coluna++) {
        se (coluna < 3) {
            escreva(S[linha][coluna] + " ")
        }
        senao {
            escreva(S[linha][coluna] + "\n")
        }
    }
}
}
```