

Lista de Exercícios - Estrutura de Dados Homogênea Unidimensional - Vetor (Parte I)

1. Escreva um algoritmo que leia o nome e idade de 15 pessoas. Em seguida o algoritmo deve informar os nomes das pessoas com idade acima da média das idades.
2. Faça um algoritmo que leia 15 números inteiros e preencha um vetor. Após a leitura dos 15 números o algoritmo deve informar:
 - O conteúdo do vetor em ordem inversa a da leitura
 - O primeiro e último número armazenados no vetor
 - O somatório dos números armazenados no vetor
 - A média aritmética dos números do vetor
3. Faça um algoritmo preencha um vetor de 30 posições, onde cada elemento corresponde ao quadrado de sua posição. Em seguida exiba os valores do vetor.
4. Escreva um algoritmo que preencha um vetor com 15 números inteiros. Em seguida um outro vetor deve ser preenchido obedecendo a regra: “todo elemento do segundo vetor deve ser o quadrado do elemento correspondente do primeiro vetor”. Os elementos dos dois vetores devem ser exibidos no final.
5. Elabore um algoritmo que leia números inteiros e preencha um vetor de tamanho 10. Em seguida exiba os valores localizados nas posições pares.
6. Faça um algoritmo que leia 20 números reais e preencha um vetor. Modifique o vetor de modo que os valores das posições ímpares sejam aumentados em 5%, e os das posições pares sejam aumentados em 2%. Em seguida exiba os valores do vetor
7. Elabore um algoritmo que leia números inteiros e preencha um vetor de tamanho 5. Em seguida crie e exiba um segundo vetor na ordem inversa do primeiro.
8. Faça um algoritmo que leia 10 números inteiros e armazene em um vetor A. Leia outros 10 números e armazene em um vetor B. Gere um terceiro vetor C onde cada elemento seja a soma dos outros dois elementos dos vetores A e B. Por exemplo:

A					
11	20	33	47	60	...
B					
2	3	2	5	6	...
C					
13	23	35	52	66	...

9. Escreva um programa que leia e armazene 10 números (inteiros) em dois vetores (A e B), ou seja, cada vetor vai armazenar 10 números inteiros. O algoritmo deve informar quais números do primeiro vetor (A) estão repetidos no segundo vetor (B). Por exemplo:
 - Vetor A: 2, 3, 44, 5, 6, 10, 45, 4, 7, 8
 - Vetor B: 1, 5, 33, 3, 8, 45, 41, 7, 9, 0
 - Números repetidos: 3, 5, 45, 7, 8
10. Faça um algoritmo que preencha um vetor com 15 números inteiros, mas o vetor não pode armazenar números repetidos. Os elementos do vetor devem ser exibidos no final.
11. Elabore um algoritmo que receba, inicialmente, o valor de uma aplicação e de uma taxa de juros. Considerando que essa taxa de juros aumenta na razão de 0,025% ao mês, o algoritmo deve armazenar em vetores, com 12 elementos cada, o valor das taxas de juros, o valor dos juros e o valor da aplicação corrigida. Mostre o valor inicial da aplicação e o conteúdo dos vetores.

Lista de Exercícios - Estrutura de Dados Homogênea Unidimensional - Vetor (Parte II)

12. Faça o rastreo do algoritmo abaixo e registre o que acontece na memória e tela do computador.

```
algoritmo "exercicio"
var
numeros: vetor [1..10] de inteiro
posicao, soma : inteiro
inicio
soma <- 0
para posicao de 1 ate 10 passo 1 faca
    escreva("Informe um número: ")
    leia(numeros[posicao])
    soma <- soma + numeros[posicao]
fimpara
para posicao de 2 ate 10 passo 2 faca
    numeros[posicao] <- numeros[posicao] + soma
fimpara
para posicao de 1 ate 10 passo 1 faca
    escreval(numeros[posicao])
fimpara
fimalgoritmo
```

13. Faça o rastreo do algoritmo abaixo e registre o que acontece na memória e tela do computador.

```
algoritmo "exercicio"
var
vet1: vetor [1..10] de inteiro
vet2: vetor [1..10] de inteiro
posicao, posicaoAuxiliar, soma : inteiro
inicio
soma <- 0
para posicao de 1 ate 10 passo 1 faca
    escreva("Informe um número: ")
    leia(vet1[posicao])
    soma <- soma + vet1[posicao]
fimpara
posicaoAuxiliar <- 10
para posicao de 1 ate 10 passo 1 faca
    vet2[posicao] <- vet1[posicaoAuxiliar] + soma
    posicaoAuxiliar <- posicaoAuxiliar - 1
fimpara
escreval("Valores do primeiro vetor: ")
para posicao de 1 ate 10 passo 1 faca
    escreval(vet1[posicao])
fimpara
escreval("Soma dos valores do primeiro vetor: ", soma)
escreval("Valores do segundo vetor")
para posicao de 1 ate 10 passo 1 faca
    escreval(vet2[posicao])
fimpara
fimalgoritmo
```

14. Faça o rastreamento do algoritmo abaixo e registre o que acontece na memória e tela do computador.

```
algoritmo "exercício"
var
    numeros: vetor [1..10] de inteiro
    posicao, valorAuxiliar, posicaoAuxiliar : inteiro
inicio
    para posicao de 1 ate 10 passo 1 faca
        escreva("Informe um número: ")
        leia(numeros[posicao])
    fimpara
    posicaoAuxiliar <- 10
    para posicao de 1 ate 5 passo 1 faca
        valorAuxiliar <- numeros[posicao]
        numeros[posicao] <- numeros[posicaoAuxiliar]
        numeros[posicaoAuxiliar] <- valorAuxiliar
        posicaoAuxiliar <- posicaoAuxiliar - 1
    fimpara
    para posicao de 1 ate 10 passo 1 faca
        escreval(numeros[posicao])
    fimpara
fimalgoritmo
```

15. Faça o rastreamento do algoritmo abaixo e registre o que acontece na memória e tela do computador.

```
algoritmo "exercício"
var
    numeros: vetor [1..10] de inteiro
    posicao, valor : inteiro
    condicao : logico
inicio
    para posicao de 1 ate 10 passo 1 faca
        condicao <- verdadeiro
        enquanto condicao faca
            escreva("Informe um número: ")
            leia(valor)
            se ((posicao mod 2) = 0) entao
                se ((valor mod 2) = 0) entao
                    condicao <- falso
                fimse
            senao
                se ((valor mod 2) <> 0) entao
                    condicao <- falso
                fimse
            fimse
        fimenquanto
        numeros[posicao] <- valor
    fimpara
    para posicao de 1 ate 10 passo 1 faca
        escreval(numeros[posicao])
    fimpara
fimalgoritmo
```