

- 1- Crie um algoritmo que crie um vetor de 10 posições, insira os números de 1 a 10. Depois imprima os valores.
- 2- Crie um algoritmo com 30 posições, peça que o usuário preencha com valores aleatórios. Após preenchido identifique o maior e o menor elemento do array.
- 3- Crie um algoritmo que leia 50 valores e insira em um array. Posteriormente, calcule a média aritmética dos valores do array.
- 4- Crie uma função que você passa um array e uma posição por parâmetro e ele retorna o elemento da posição. Se posição inválida, informar ao usuário: "Posição inválida".
- 5- Escreva programa que possua um vetor denominado A que armazene 6 números inteiros. O programa deve executar os seguintes passos
 - Atribua os seguintes valores a esse vetor: 1, 0, 5, -2, -5, 7.
 - Armazene em uma variável inteira a soma entre os valores das posições A[0], A[1] e A[5] do vetor e mostre na tela esta soma.
 - Modifique o vetor na posição 4, atribuindo a esta posição o valor 100.
 - Mostre na tela cada valor do vetorA, um em cada linha.
- 6- Escreva um algoritmo que leia um vetor de 15 posições e, em seguida, leia também dois valores X e Y quaisquer correspondentes a duas posições no vetor. Ao final seu programa deverá escrever a soma dos valores encontrados nas respectivas posições X e Y. Se X ou Y forem posições inválidas, informar o usuário "Posições inválidas", pedir uma posição válidas novamente.
- 7- Fazer um programa para ler 8 valores e, em seguida, mostrar a posição onde se encontram o maior e o menor valor.
- 8- Crie um algoritmo que recebe um array de inteiros X e um valor inteiro Y e retorna a quantidade de vezes que Y aparece no array X
- 9- Crie um algoritmo que além da função main, tenha mais 3 funções que receba um vetor por parâmetro e cada função deve retornar respectivamente o maior valor, menor valor e o valor que se encontra na metade do array. Tanto o tamanho do vetor quanto todos os elementos inseridos devem ser informados pelo usuário.
- 10- Foi realizada uma pesquisa de algumas características físicas de 50 habitantes de uma certa região. De cada habitante foram coletados os seguintes dados: gênero, cor dos olhos (azuis, verdes ou castanhos), cor dos cabelos (louros, pretos ou castanhos) e idade. Faça um algoritmo que crie e leia do usuário os seguintes valores e insira em cada respectivo vetor:

- genero: tipo char que armazena 'f' para feminino e 'm' para masculino
- olhos: tipo int, no qual 1- azul, 2- verde 3- castanho
- cabelos: tipo int, no qual 1- louro, 2- preto, 3- castanho

No final, a saída deve informar a quantidade de cada item.

- Exemplo:
 - Gênero:
 - 21 homens
 - 29 mulheres
 - Olhos:
 - 10 azuis
 - 15 verdes
 - 25 castanhos
 - Cabelos:
 - 10 louros
 - 20 pretos
 - 20 castanhos

11- Crie uma função que receba dois vetores A e B como parâmetros e no final você deve retornar o vetor C resultado da união entre A e B (operação de conjunto)

12- Crie uma função que receba dois vetores A e B como parâmetros e no final você deve retornar o vetor C resultado da intersecção entre A e B (operação de conjunto)

13- Crie uma função que receba dois vetores A e B como parâmetros e no final você deve retornar o vetor C resultado da diferença entre A e B (operação de conjunto)

14- Escreva um algoritmo que leia um vetor de 15 posições e o compacte, ou seja, elimine as posições com valor zero. Para isso, todos os elementos à frente do valor zero, devem ser movidos uma posição para trás no vetor.

15- Escreva um algoritmo para ler 10 números DIFERENTES (seu programa tem de garantir que não existem números iguais no array) a serem armazenados em um vetor. Os dados deverão ser armazenados no vetor na ordem que serão lidos, sendo que caso o usuário digite um número que já foi digitado anteriormente, o programa deverá pedir para ele digitar outro número. Note que cada valor digitado pelo usuário deve ser pesquisado no vetor, verificando se ele existe entre os números que já foram fornecidos. Exibir na tela o vetor final que foi digitado.

16- Escreva um algoritmo que crie um vetor de inteiros de 15 posições e leia cada elemento do usuário, os elementos precisam estar ordenados de forma crescente. Ou seja, toda vez que informar um valor é necessário varrer o vetor procurando a posição ideal para inserí-lo. E "empurrar" os elementos para frente.