

### **Lista de Exercícios 08**

Desenvolva os exercícios abaixo utilizando somente o que foi visto em sala de aula. Novas soluções são encorajadas, no entanto, é necessário que os alunos demonstrem domínio sobre as técnicas apresentadas em sala de aula.

1. Dada uma seqüência de  $n$  números, imprimi-la na ordem inversa à da leitura.
2. Tentando descobrir se um dado era viciado, um dono de cassino honesto (ha! ha! ha! ha!) o lançou  $n$  vezes. Dados os  $n$  resultados dos lançamentos, determinar o número de ocorrências de cada face.
3. Faça um programa onde você declara um vetor de 10 posições que receba respectivamente 10 valores do tipo float (você deverá digitá-los via teclado). Apresente então, a média desses 10 valores.
4. Faça um programa que leia 10 valores do teclado, e que imprima esses valores divididos em 2 etapas: primeiro, somente os valores pares, e depois, somente os valores ímpares. Identifique cada valor escrevendo ao lado do valor se ele é par ou ímpar.
5. Fazer um algoritmo que leia vinte números inteiros armazenando-os em um vetor, a seguir, troque o conteúdo da primeira posição do vetor com o conteúdo da décima primeira posição, o conteúdo da segunda posição com o da décima segunda, e assim, sucessivamente até trocar o conteúdo da décima com o da vigésima e escreva o vetor após estas modificações.
6. Escreva uma palavra, letra por letra, colocando cada letra em um vetor. Ao final, imprima na tela a palavra digitada, com o formato de uma palavra (uma letra após a outra).

7. Faça um programa, utilizando vetores, que armazene o resultado de uma votação. Considere que você tem 5 candidatos, e para cada voto, você deverá digitar 1, 2, 3, 4 ou 5, que correspondem aos códigos de cada candidato. O programa deve parar de receber votos quando for digitado -1. Outros valores devem ser ignorados. No final, indique qual o candidato vencedor da votação.
8. Escrever um programa que declare um vetor de 20 inteiros, leia um valor para cada posição e no final mostre quantos elementos possuem valor maior, menor e igual ao elemento do vetor que o usuário escolher
9. Faça um programa que começa com um array pré-definido com 5 valores, o usuário deve acertar os 5 valores durante a execução do programa. O programa deve terminar quando todos os valores forem acertados. Dica: utilize um array extra.
10. Crie 2 vetores de 10 posições cada um (chame de `v1[10]` e `v2[10]`). Digite 10 valores para `v1`, usando `printf` e `scanf`. Agora, use o `v2` para armazenar A SOMA de TODOS os valores de índices anteriores de `v1`, inclusive o próprio índice. Exemplo:  
`v2[0]` deve armazenar o valor de `v1[0]`;  
`v2[1]` deve armazenar o valor de `v1[0] + v1[1]`;  
`v2[2]` deve armazenar o valor de `v1[0] + v1[1] + v1[2]`;
11. Faça um programa que recebe dois vetores de tamanho que o usuário escolha. Execute a soma destes, elemento a elemento e coloque em um outro vetor.
12. Guarde em um array 10 idades. Exiba quantas pessoas são maior de idade (18 anos) e quantas são menores.
13. Leia um vetor de 12 posições e em seguida ler 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. O programa deve continuar rodando e perguntando novas posições até que o usuário digite -1 como posição.

14. Crie um vetor com 10 números e, dentro de um laço de repetição permita que o usuário escolha duas posições e troque o valor destas posições. A cada troca o programa deve mostrar o vetor em questão.

15. Leia um vetor de 20 posições e em seguida um valor X qualquer. Seu programa deverá fazer uma busca do valor de X no vetor lido e informar a posição em que foi encontrado ou se não foi encontrado.

Desafio 1. Dado dois vetores de tamanho 5 e com valores ordenados, criar um vetor de tamanho 10 com todos elementos ordenados. O programa deve receber 5 valores do primeiro vetor, 5 valores do segundo vetor e, finalmente, mostrar o vetor final com 10 posições com os valores ordenados. Ex:  $v1 = [1, 3, 5, 7, 9]$  e  $v2 = [2, 4, 6, 8, 10]$ . O programa deverá mostrar o vetor  $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

Desafio 2. Pesquise e implemente o algoritmo de ordenação conhecido como Bubble-sort