

**Instruções:**

- Leia todas as instruções dos enunciados corretamente para poder desenvolver seus programas;
- Evite plágio (será verificado por meio de ferramentas automatizadas). Faça seu programa com os seus nomes de variáveis e lógica de solução. Plágios identificados anularão as atividades entregues de todos os envolvidos.
- Adicione comentários nos códigos explicando seu raciocínio e sua tomada de decisão. Porém, não exagere nos comentários, pois a própria estrutura do programa deve ser auto-explicativa.
- crie um arquivo com extensão .c para cada exercício. Por exemplo, o exercício 01 pode ser gerado em um arquivo **ex01.c**; A lista pode ser resolvida individualmente ou em dupla.
- o envio da lista será feito pelo Moodle. Gere um arquivo compactado com todos os programas desenvolvidos e adicione a identificação (nomes) dos alunos envolvidos na resolução.

---

## 1 Categoria de Exercícios: Vetores

---

**Exercício 1.** Seja um vetor inteiro de 9 elementos. Escreva um programa que realize a seguinte sequência de ações na ordem indicada:

- preenchimento do vetor por leitura, sendo aceitos para armazenamento apenas valores positivos, maiores do que zero;
- impressão dos elementos do vetor em uma linha, usando dois espaços brancos para separar os valores de cada elemento;
- rotação dos elementos do vetor, com o deslocamento de todos os valores dos elementos para a posição seguinte, exceto o último valor, que deve ser colocado na posição do primeiro elemento;
- repetir o item b);
- soma do índice de cada elemento ao conteúdo da posição correspondente;
- repetição do item b);
- leitura do valor inteiro. Determinação e apresentação do número de ocorrências desse valor no vetor.

**Exercício 2.** Faça um programa que preencha, por leitura, um vetor inteiro de 20 elementos e determine quantos elementos diferentes existem nesse vetor. Se, por exemplo, o vetor contiver 10 vezes o valor 1 e 20 vezes o valor -1, a resposta do programa será 2.

**Exercício 3.** Faça um programa que preencha, por leitura, com valores reais, dois vetores A e B, ambos de oito elementos, e realize a troca dos elementos desses vetores. Após a execução do programa, o vetor B deverá conter os valores fornecidos para o vetor A e vice-versa.

**Exercício 4.** Faça um programa que leia os conteúdos de dois vetores inteiros (X e Y) com 10 elementos cada e os imprima. O programa deverá, a seguir:

- preencher outro vetor U, de forma que esse vetor seja a união de X com Y;
- preencher outro vetor I, de forma que esse vetor seja a intersecção de X e Y;

c) imprimir os dois vetores U e I.

**Exercício 5.** Escreva um programa que preencha, por leitura, um vetor **vet** inteiro com 20 elementos, e em seguida, imprima os valores lidos. O programa deverá gerar, a partir do vetor **vet**, um vetor **pos** apenas com os valores positivos maiores que zero. A partir do vetor **pos**, deverá ser gerado ainda outro vetor, **semdup**, em que haja apenas uma ocorrência de cada valor existente em **pos**. Tanto o vetor **pos**, como o vetor **semdup**, devem ser gerados de modo contínuo, da primeira posição em diante, ou seja, eventuais posições não ocupadas poderão existir apenas no final dos vetores. Ao final do processamento, apresentar os três vetores. No caso dos vetores **semdup** e **pos**, apresentar apenas suas posições efetivamente utilizadas.

**Exercício 6.** Escreva um algoritmo de pesquisa sequencial que informa a posição em que se encontra o valor buscado. Caso esse valor não seja encontrado, informe a posição -1.

**Exercício 7.** Faça um programa que calcule uma média móvel. O programa deverá executar um laço de leitura de valores inteiros e positivos. A introdução de valores negativos servirá como indicador de término do programa. Para cada valor fornecido, deverá ser impressa a média calculada. A média móvel é calculada sobre um número especificado de pontos. Quando se introduz um novo dado, o valor mais antigo é descartado, e o novo valor introduzido é incorporado ao cálculo da média. Esse esquema de substituição faz da média nível um instrumento valioso na análise de tendências. Considerar para a solução deste problema 5 pontos (valores) e iniciar o cálculo das médias ao completar a quinta leitura.

**Exercício 8.** Faça um programa para o controle do estoque de uma loja. A loja vende 10 produtos diferentes. O programa deve iniciar lendo os códigos de cada um desses produtos e a quantidade de itens dos mesmos existente no estoque da loja, armazenando esses valores em dois vetores, um para os códigos e outro para as quantidade de itens (mesmos índices nos dois arranjos devem corresponder ao mesmo produto). O programa deve processar um conjunto de atualizações de estoque, com inserção e retirada de itens. No final do processo, o programa deve fazer uma análise do estoque que restou na loja, informando:

- a) códigos dos produtos que estão com estoque inferior a 10 unidades;
- b) número de produtos que apresentam estoque entre 10 e 20 unidades (inclusive); e
- c) número total de itens em estoque para cada produto.

**Exercício 9.** Uma lanchonete vende cinco sanduíches diferentes, com os preços indicados na tabela a seguir: Faça um programa que processe diversos pedidos. Em cada pedido, poderão ser solicitadas quantidade

Código	Produto	Preço Unitário (R\$)
1	Sanduíche aberto	12,00
2	Sanduíche universitário	3,00
3	Misto quente	4,00
4	Queijo quente	3,50
5	Bauru	6,00

variadas dos vários tipos de sanduíches. Leia os preços dos sanduíches no início do programa e armazene-os em um vetor, cujos índices correspondem os códigos dos sanduíches. Armazene o total de itens solicitados de cada tipo de sanduíche em outro vetor. No final de cada pedido, deve ser calculado e informado o valor total a pagar, além do número total de cada um dos itens solicitados. No final do dia, o programa deverá informar:

- número de sanduíches de cada tipo vendidos;
- valor total (em reais) vendido no dia;
- número de clientes atendidos;
- valor do pedido médio.

---

**Exercício 10.** Uma empresa vende 10 artigos diferentes. Os artigos possuem preços variados, sendo cada um identificado por um código único. Usando vetores para armazenar estas informações, escreva um programa que:

- a) obtenha os códigos e os preços dos 10 artigos e imprima essas informações;
  - b) identifique e informe o código e o preço dos três artigos mais caros;
  - c) calcule e informe a média dos preços de todos os artigos;
  - d) informe os códigos dos artigos com preço superior à média.
- 

## 2 Categoria de Exercícios: Strings

---

**Exercício 11.** Construa um programa que leia através da entrada padrão uma *string* e retorne na saída padrão o número de caracteres que a mesma possui.

**Exercício 12.** Faça um programa que leia uma *string* e conte a quantidade de vogais.

**Exercício 13.** Faça um programa que receba uma palavra e imprima uma nova palavra sendo cada letra a seguinte a da palavra original. Por exemplo: Banana → Cbobob.

**Exercício 14.** Faça um programa que receba uma palavra com todas as letras minúsculas e transforme-as em maiúscula. Exemplo: banana → BANANA.

**Exercício 15.** Escreva um programa que leia uma senha alfanumérica. Utilize a função `strcmp()` da biblioteca `string.h` para compará-la com uma senha definida internamente no programa e retorne ao usuário a validade ou não da senha fornecida por ele, em função do resultado da comparação.

**Exercício 16.** Leia uma *string* de tamanho qualquer e indique qual é o caractere que mais aparece, e quantas vezes ele ocorre nesta *string*. Por exemplo:

Entrada: Vamos estudar strings

Saída: O carácter que mais aparece é s. Apareceu 4 vezes.

**Obs:** Se existirem 2 ou mais caracteres de maior ocorrência, todos eles deverão ser mostrados.

**Exercício 17.** Escreva um programa que receba uma *string* e coloque em maiúsculo a primeira letra de cada palavra dessa string. Exemplo:

Entrada: abobrinha com feijao, muito bom, ou nao.

Saída: Abobrinha Com Feijao, Muito Bom, ou Nao.

**Exercício 18.** Escreva um programa que receba duas *strings*. As duas *strings* são informadas pelo usuário. A primeira corresponde a um texto. A segunda é uma *string* qualquer. Verifique se a segunda está contida no texto fornecido pelo usuário.

**Exercício 19.** Faça um programa que receba uma *string*, a qual corresponda ao CPF de uma pessoa. Valida o CPF. Procure na Internet pelas regras de validação.

**Exercício 20.** Faça um jogo da forca.