

## Exercícios 08

8.1 a) Escreva uma função chamada **escreveVetor** que receba como entrada um vetor de inteiros, um inteiro N representando a quantidade de elementos existentes no vetor e exiba o vetor conforme o layout abaixo. Neste exemplo considere um vetor com três elementos: **[10,20,40]**

**Parâmetros de entrada:** Um vetor de inteiros e quantidade de elementos existentes no vetor.

**Retorno:** Nenhum.

b) Escreva uma função chamada **leVetor** para ler um vetor de N elementos. Os dados lidos devem ser armazenadas em um vetor passado como saída. Considere que N é um parâmetro de entrada.

**Parâmetros de saída:** Um vetor de inteiros

**Parâmetro de entrada:** quantidade de elementos existentes no vetor.

**Retorno:** Nenhum.

c) Escreva uma função chamada **multiplica** que receba como entrada um vetor de inteiros, e dois inteiros N e A. O inteiro N representa a quantidade de elementos do vetor de entrada e A o inteiro que multiplicará cada elemento do vetor de entrada. A função deve armazenar em um vetor de saída os elementos do vetor de entrada multiplicados por A.

**Parâmetro de entrada:** Um vetor e dois inteiros N e A.

**Parâmetros de saída:** O vetor que armazena o resultado da multiplicação.

**Retorno:** Nenhum.

d) Escreva um programa em C que faça o seguinte: Lê um inteiro N (máximo 10). A seguir ler um vetor de N elementos. Logo após ler vários inteiros. Multiplicar cada inteiro lido pelo vetor de entrada. Exibir o vetor resultante de cada multiplicação. O programa termina ao ser informado um valor 0 para o inteiro, nesta situação nenhuma multiplicação deve ser executada. Utilizar as funções **leVetor**, **escreveVetor** e **mutiplica** para implementar o programa.

8.2 a) Escreva uma função chamada **obtemMaior** que recebe como entrada um vetor e um inteiro N representando a quantidade de elementos do vetor. A função deve retornar o maior elemento armazenado no vetor de entrada.

**Parâmetro de entrada:** Um vetor e um inteiro N.

**Retorno:** O maior elemento armazenado no vetor.

b) Escreva um programa em C que leia um inteiro N. A seguir ler um vetor de N elementos (utilizar a função **leVetor**). Exibir o maior elemento armazenado no vetor. Utilizar a função **obtemMaior**.

8.3 a) Escreva uma função chamada **calculaSoma** que recebe como entrada um vetor e três inteiros N, A e B. N representa a quantidade de elementos existente no vetor de entrada. A e B índices que delimitam um intervalo dentro do vetor (considere que são válidos e que A pode ser menor que B e vice-versa). A função deve retornar a soma dos valores do vetor que estão localizados entre os índices passados como parâmetro (incluir na soma os elementos armazenados nos índices informados).

**Parâmetro de entrada:** Um vetor e três inteiros N, A e B.

**Retorno:** A soma conforme descrito no enunciado.

b) Escreva um programa em C para ler um valor N (máximo 10). A seguir ler um vetor de N elementos (utilizar a função **leVetor**). Depois ler várias duplas de inteiros representando índices de um vetor. Para cada dupla escrever a soma dos elementos armazenados no vetor contidos entre (inclusive) os índices informados (Utilizar a função **calculaSoma**). O programa termina a serem informados valores iguais. Nesta situação a soma também deve ser impressa.

8.4 a) Escreva um função chamada **obtemMajiores** que recebe como entrada um vetor e dois inteiros N e A. A função deve copiar para outro vetor (de saída) os elementos armazenados no vetor de entrada que são maiores que A (sem deixar espaços em branco entre os elementos do vetor de saída). A função deve retornar a quantidade de elementos copiados.

**Parâmetro de entrada:** Um vetor e dois inteiros N e A.

**Parâmetro de saída:** Um vetor com os elementos maiores que A que estão armazenados no vetor de entrada.

**Retorno:** Quantidade de elementos copiados no vetor de saída.

b) Escreva um programa em C para ler um valor N (máximo 10). A seguir ler um vetor de N elementos (utilizar a função **leVetor**). Depois ler vários inteiros. Para cada inteiro informado imprimir os elementos do vetor lido que são maiores que ele. Obter os maiores com a chamada da função **obtemMajiores**. Caso não exista no vetor pelo menos um elemento maior que o inteiro lido escrever a mensagem **“Não há elementos”**. O programa termina ao ser informado um inteiro negativo. Nesta situação nenhum valor deve ser impresso.

**8.5 a)** Escreva uma função chamada **rotacionaDir** que recebe como entrada um vetor e um inteiro N (quantidade de elementos armazenados no vetor). A função deve rotacionar os elementos do vetor para a direita de forma que o último elemento passe a ser o primeiro.

**Parâmetro de entrada/saída:** Um vetor.

**Parâmetro de entrada:** Um inteiro.

**Retorno:** Nenhum

**b)** Escreva uma função chamada **rotacionaEsq** que recebe como entrada um vetor e um inteiro N (quantidade de elementos armazenados no vetor). A função deve rotacionar os elementos do vetor para a esquerda de forma que o primeiro elemento passe a ser o último.

**Parâmetro de entrada/saída:** Um vetor.

**Parâmetro de entrada:** Um inteiro.

**Retorno:** Nenhum

**c)** Escreva um programa em C para ler um valor N (máximo 10). A seguir ler um vetor de N elementos (utilizar a função **leVetor**). A seguir ler vários códigos (**1.Direita 2.Esquerda 3.Fim**) que representam a direção da rotação que deve ser executada sobre o vetor de entrada. Para cada código informado executar a respectiva rotação e exibir o vetor (utilizar a função **escreveVetor**). O programa termina ao ser informado o código 3. Para executar as rotações utilizar as funções **rotacionaDir** e **rotacionaEsq**.

Tabela de dados para os programas.

As mensagens entre parênteses apenas indicam o significados dos dados.

#### 8.1)

[Entrada]	[Saída]
5 (N)	
4 8 10 4 3 (vetor)	
3	[12 24 30 12 9] (vetor de saída)
1	[4 8 10 4 3] (vetor de saída)
2	[8 16 20 8 6] (vetor de saída)
0	

#### 8.2)

[Entrada]	[Saída]
5 (N)	
4 8 10 4 3 (vetor)	10 (maior elemento)

#### 8.3)

[Entrada]	[Saída]
8 (N)	
4 8 10 4 3 2 1 6 (vetor)	
0 2	22 (soma)
2 0	22 (soma)
3 6	10 (soma)
1 1	8 (soma)

#### 8.4)

[Entrada]	[Saída]
8 (N)	
4 8 10 4 3 1 2 6 (vetor)	
5	[8 10 6]
9	[10]
20	Não existe
3	[4 8 10 4 6]
-1	

#### 8.5)

[Entrada]	[Saída]
8 (N)	
4 8 10 4 3 1 2 6 (vetor)	
1 (direita)	[6 4 8 10 4 3 1 2]
1 (direita)	[2 6 4 8 10 4 3 1]
2 (esquerda)	[6 4 8 10 4 3 1 2]
3 (fim)	