



Exercícios – Funções e Vetores

Funções Utilitárias:

Implemente funções utilitárias que devem ser empregadas como parte das soluções dos demais exercícios nesta lista. A primeira delas deve ler do usuário os elementos de um vetor. Para um vetor de inteiros, uma possível assinatura poderia ser:

```
void le_vetor(int v[], int n);
```

A segunda função utilitária deve escrever todos os elementos do vetor na tela. Para um vetor de inteiros, uma possível assinatura poderia ser:

```
void imprime_vet(int v[], int n);
```

Questões:

1. Implemente uma função que receba como parâmetro de entrada um vetor de números reais. A função a ser implementada deve retornar a média dos valores no vetor. A função **main** deve ler o tamanho **n** do vetor, cada um dos seus elementos e exibir na tela uma mensagem informando a média dos valores do vetor utilizando a função implementada. Exemplo:

```
Informe o tamanho do vetor:
5
Informe os elementos do vetor:
2.5 3.5 5.0 9.0 1.0
Media do vetor: 4.2
```

2. Implemente uma função que receba como parâmetro de entrada um vetor de números reais. A função a ser implementada deve retornar a quantidade de valores que são menores do que a média entre eles, utilizando para isto uma chamada à função da Questão 1. A função **main** deve ler o tamanho **n** do vetor, cada um dos seus elementos e exibir na tela uma mensagem informando quantos elementos do vetor são maiores do que a média utilizando a função implementada. Exemplo:

```
Informe o tamanho do vetor:
5
Informe os elementos do vetor:
2.5 3.5 5.0 9.0 1.0
Elementos menores do que a media: 3
```

3. Implemente uma função que receba como parâmetro de entrada um vetor de números inteiros. A função a ser implementada deve retornar o maior elemento do vetor. A função **main** deve ler o tamanho **n** do vetor, cada um dos seus elementos e exibir na tela uma mensagem informando todas as posições do vetor em que o maior elemento está presente utilizando a função implementada. Exemplo:

Informe o tamanho do vetor:

6

Informe os elementos do vetor:

-11 7 1 2 7 7

Maior elemento ocorre nas posicoes: 1 4 5

4. Implemente uma versão recursiva para a função que retorna o maior elemento de um vetor.
5. Implemente uma função que receba como parâmetros de entrada dois vetores de números reais. A função a ser implementada deve retornar o produto interno entre os dois vetores, ou seja,

$$\mathbf{u}^t \cdot \mathbf{v} = \sum_{i=0}^{n-1} u_i \cdot v_i = u_0 \cdot v_0 + u_1 \cdot v_1 + \dots + u_{n-1} \cdot v_{n-1},$$

com

$$\mathbf{u} = [u_0, u_1, \dots, u_{n-1}]^t$$

e

$$\mathbf{v} = [v_0, v_1, \dots, v_{n-1}]^t.$$

A função `main` deve ler o tamanho `n` dos vetores, cada um dos elementos de ambos os vetores e, utilizando chamadas à função implementada, exibir na tela mensagens informando:

- (a) O produto interno entre os dois vetores
- (b) A norma do primeiro vetor
- (c) A norma do segundo vetor

Observe que a norma $\|\mathbf{u}\|$ de um vetor qualquer pode ser obtida pela relação

$$\|\mathbf{u}\|^2 = \mathbf{u}^t \cdot \mathbf{u}$$

Exemplo:

Informe o tamanho dos vetores:

3

Informe os elementos do vetor1:

1.0 0.0 2.0

Informe os elementos do vetor2:

0.0 1.0 1.0

Produto interno: 2

Norma do vetor1: 2.23607

Norma do vetor2: 1.41421

6. Implemente uma função que receba como parâmetros de entrada dois vetores de números inteiros e como parâmetro de saída um outro vetor de números inteiros. A função a ser implementada deve armazenar no vetor de saída a concatenação de todos os valores do primeiro vetor seguidos por todos os valores do segundo vetor, nesta ordem. A função `main` deve ler o tamanho de cada vetor (eles podem ter tamanhos diferentes), cada um dos elementos de ambos os vetores e, utilizando uma chamada à função implementada, exibir o vetor resultante. Exemplo:

Informe o tamanho do primeiro vetor:

4

Informe o tamanho do segundo vetor:

3

Informe os elementos do vetor1:

1 -5 8 7

Informe os elementos do vetor2:

1 5 3

Vetor resultante:

1 -5 8 7 1 5 3

7. Implemente uma função que receba como parâmetros de entrada dois vetores de números inteiros e como parâmetro de saída um outro vetor de números inteiros. A função a ser implementada deve armazenar no vetor de saída o entrelaçamento dos valores do primeiro vetor com os valores do segundo vetor (primeiro elemento do primeiro vetor, primeiro elemento do segundo vetor, segundo elemento do primeiro vetor, segundo elemento do segundo vetor, etc.). A função `main` deve ler o tamanho de cada vetor (eles podem ter tamanhos diferentes), cada um dos elementos de ambos os vetores e, utilizando uma chamada à função implementada, exibir o vetor resultante. Exemplo:

Informe o tamanho do primeiro vetor:

4

Informe o tamanho do segundo vetor:

3

Informe os elementos do vetor1:

1 -5 8 7

Informe os elementos do vetor2:

1 5 3

Vetor resultante:

1 1 -5 5 8 3 7

8. Implemente uma função que receba como parâmetro de entrada um vetor de inteiros e como parâmetros de saída outros dois vetores de inteiros. A função a ser implementada deve armazenar no primeiro vetor de saída todos os números pares e no segundo vetor de saída todos os números ímpares. Observe que esta função também deve computar o tamanho dos vetores de saída, ou seja, a quantidade de números pares e ímpares: para isto, faça com que o tamanho de cada vetor de saída seja também um parâmetro de saída. A função `main` deve ler o tamanho do vetor, cada um dos seus elementos e, utilizando uma chamada à função implementada, exibir os vetores resultantes. Exemplo:

Informe o tamanho do vetor:

7

Informe os elementos do vetor:

0 1 2 3 4 5 6

Elementos pares:

0 2 4 6

Elementos ímpares:

1 3 5

9. Implemente uma função que receba como parâmetro de entrada um número inteiro e como parâmetro de saída um vetor de inteiros. A função a ser implementada deve armazenar no vetor cada bit da conversão do número decimal para binário. A conversão de um número decimal para binário é dada por sucessivas

divisões do número de entrada por 2 até que o quociente seja 0 ou 1, sendo cada bit composto pelo resto da divisão por 2. Quando o quociente é igual a 0 ou 1, este se torna o bit mais significativo (ou seja, o bit mais à esquerda). Por exemplo, a conversão de 13 para binário é igual ao vetor de bits [1, 1, 0, 1], o qual é computador por:

```
13 % 2 = 1
      (13/2 = 6)
6 % 2 = 0
      (6/2 = 3)
3 % 2 = 1
3/2 = 1
```

A função `main` deve ler apenas o número decimal a ser convertido. A partir deste valor, o programa deve computar a quantidade de bits (ou seja, o tamanho `n`) do vetor e utilizando uma chamada à função implementada, exibir na tela o número binário resultante. Exemplo:

```
Informe um numero decimal:
16
Numero binario:
1 0 0 0 0
```