

# **Algoritmos e Estruturas de Dados I**

## **Sexta Lista de Exercícios - Ponteiros**

1. Faça um programa que leia dois valores inteiros e chame uma função que receba estes 2 valores de entrada e retorne o maior valor na primeira variável e o menor valor na segunda variável. Escreva o conteúdo das 2 variáveis na tela.
2. Elaborar um programa que leia dois valores inteiros (A e B). Em seguida faça uma função que retorne a soma do dobro dos dois números lidos. A função deverá armazenar o dobro de A na própria variável A e o dobro de B na própria variável B.
3. Crie um programa que contenha uma função que permita passar por parâmetro dois números inteiros A e B. A função deverá calcular a soma entre estes dois números e armazenar o resultado na variável A. Esta função não deverá possuir retorno, mas deverá modificar o valor do primeiro parâmetro. Imprima os valores de A e B na função principal.
4. Crie um programa que contenha um array de float contendo 10 elementos. Imprima o endereço de cada posição desse array.
5. Crie um programa que contenha uma matriz de float contendo 3 linhas e 3 colunas. Imprima o endereço de cada posição dessa matriz.
6. Crie um programa que contenha um array de inteiros contendo 5 elementos. Utilizando apenas aritmética de ponteiros, leia esse array do teclado e imprima o dobro de cada valor lido.
7. Crie um programa que contenha um array contendo 5 elementos inteiros. Leia esse array do teclado e imprima o endereço das posições contendo valores pares.
8. Faça um programa que leia três valores inteiros e chame uma função que receba estes 3 valores de entrada e retorne eles ordenados, ou seja, o menor valor na primeira variável, o segundo menor valor na variável do meio, e o maior valor na última variável. A função deve retornar o valor 1 se os três valores forem iguais e 0 se existirem valores diferentes. Exibir os valores ordenados na tela.
9. Elabore uma função que receba duas strings como parâmetros e verifique se a segunda string ocorre dentro da primeira. Use aritmética de ponteiros para acessar os caracteres das strings.
10. Crie uma função que receba dois parâmetros: um array e um valor do mesmo tipo do array. A função deverá preencher os elementos de array com esse valor. Não utilize índices para percorrer o array, apenas aritmética de ponteiros.
11. Crie uma função que receba como parâmetro um array e o imprima. Não utilize índices para percorrer o array, apenas aritmética de ponteiros.
12. Considere a seguinte declaração: `int A, *B, **C, ***D;` Escreva um programa que leia a variável A e calcule e exiba o dobro, o triplo e o quádruplo desse valor utilizando apenas os ponteiros B, C e D. O ponteiro B deve ser usada para calcular o dobro, C o triplo e D o quádruplo.

13. Escreva uma função que dado um número real passado como parâmetro, retorne a parte inteira e a parte fracionária deste número. Escreva um programa que chama esta função.

Protótipo:

```
void frac(float num, int* inteiro, float* frac);
```

14. Implemente uma função que calcule a área da superfície e o volume de uma esfera de raio R. Essa função deve obedecer ao protótipo:

```
void calc_esfera(float R, float *area, float *volume)
```

A área da superfície e o volume são dados, respectivamente, por:

$$A = 4 * \pi * R^2$$

$$V = 4/3 * \pi * R^3$$

15. Escreva uma função que aceita como parâmetro um array de inteiros com N valores, e determina o maior elemento do array e o número de vezes que este elemento ocorreu no array. Por exemplo, para um array com os seguintes elementos: 5, 2, 15, 3, 7, 15, 8, 6, 15, a função deve retornar para o programa que a chamou o valor 15 e o número 3 (indicando que o número 15 ocorreu 3 vezes). A função deve ser do tipo void.

16. Implemente uma função que receba como parâmetro um array de números reais de tamanho N e retorne quantos números negativos há nesse array. Essa função deve obedecer ao protótipo:

```
int negativos(float *vet, int N);
```

17. Escreva um programa que declare um array de inteiros e um ponteiro para inteiros. Associe o ponteiro ao array. Agora, some mais um (+1) a cada posição do array usando o ponteiro (use \*).

18. Crie uma função para somar dois arrays. Esta função deve receber dois arrays e retornar a soma em um terceiro array. Caso o tamanho do primeiro e segundo array seja diferente então a função retornará ZERO (0). Caso a função seja concluída com sucesso a mesma deve retornar o valor UM (1). Utilize aritmética de ponteiros para manipulação do array.

19. Escreva uma função que retorna o maior valor de um array de tamanho N. Escreva um programa que leia N valores inteiros, imprima o array com k elementos por linha, e o maior elemento. O valor de k também deve ser fornecido pelo usuário.

20. Escreva uma função que receba um array de inteiros V e os endereços de duas variáveis inteiras, min e max, e armazene nessas variáveis o valor mínimo e máximo do array. Escreva também uma função main que use essa função.