



## Exercícios: Ponteiros

1. Faça um programa que leia 2 valores inteiros e chame uma sub-rotina que receba estas 2 variáveis e troque o seu conteúdo, ou seja, esta rotina é chamada passando duas variáveis A e B por exemplo, e após a execução da rotina A conterá o valor de B e B terá o valor de A.
2. Faça um programa que leia dois valores inteiros e chame uma sub-rotina que receba estes 2 valores de entrada e retorne o maior valor na primeira variável e o segundo maior valor na segunda variável. Escreva o conteúdo das 2 variáveis na tela.
3. Faça um programa que leia três valores inteiros e chame uma sub-rotina 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 rotina deve retornar o valor 1 se os três valores forem iguais e 0 se existirem valores diferentes. Exibir os valores ordenados na tela.
4. Faça um programa que possua uma rotina para ler 2 notas (chamar uma rotina dedicada a ler 2 notas válidas e que devolva os 2 números lidos), outra rotina que dada estas 2 notas vá calcular a média simples e a média ponderada onde a segunda nota tem peso 2  $\rightarrow (n1 + n2 * 2)/3$ , e uma última rotina que vá exibir na tela todos os dados que foram fornecidos e calculados: rotinas `le_notas`, `calc_media` e `exibe_dados`.
5. DESAFIO: Faça um jogo no qual o computador desafia dois usuários para ver quem tem mais memória. O jogo deve gerar um número de 0 a 9 e apresentar ao jogador 1 durante 5 segundos. Depois deve repetir este procedimento para o jogador 2. Em seguida, ele deve pedir ao jogador 1 para informar o número escolhido, e após pedir ao jogador 2 para fazer a mesma coisa. Na segunda rodada, o computador deve gerar mais um número entre 0 e 9, que será 'concatenado' à direita do primeiro número. Então repetiremos o procedimento de exibição e teste de memorização para os dois usuários. O programa deve continuar adicionando números (casas adicionadas ao final do número) até que um dos dois jogadores não consiga mais memorizar perfeitamente o número. Para concluir, o programa deve informar qual dos dois jogadores ganhou, quantas casas foram memorizadas ao total, e em que casa foi que o jogador perdedor errou (primeira casa errada começando da esquerda para a direita). Dicas: procure desenvolver o programa em módulos; armazene a sequência de números dentro de uma string.
6. 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 número. Imprima os valores de A e B na função principal.
7. Implemente uma função que calcule as raízes de uma equação do segundo grau do tipo  $ax^2 + bx + c = 0$ . Essa função deve obedecer ao seguinte protótipo:

```
int raizes(float a, float b, float c, float * x1, float * x2);
```

Essa função deve ter como valor de retorno o número de raízes reais e distintas da equação. Se existirem raízes reais, seus valores devem ser armazenados nas variáveis apontadas por x1 e x2.

8. 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. Assinatura:

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

9. Escreva uma função que aceita como parâmetro um vetor de inteiros com n valores, e determina o maior elemento do vetor e o número de vezes que este elemento ocorreu no vetor. Por exemplo, para um vetor 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.
10. Crie uma função para somar dois vetores. Esta função deve receber dois vetores e retornar a soma em um terceiro vetor. Caso o tamanho do primeiro e segundo vetor seja diferente então a função retornará um erro. Caso a função seja concluída com sucesso a mesma deve retornar o valor zero (0). Utilize aritmética de ponteiros para manipulação do array. Mostre o uso dessa função em um programa feito em C.
11. Elaborar um programa em C que leia dois valores inteiros (variáveis 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
12. 12) 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:

$$A = 4 * p * R^2$$

$$V = 4/3 * p * R^3$$

13. Implemente uma função que receba como parâmetro um vetor de números reais (VET) de tamanho N e retorne quantos números negativos há nesse vetor. Essa função deve obedecer ao protótipo:

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

14. Escreva uma função que retorna o maior valor de um vetor alocado estaticamente de tamanho N. Escreva um programa que leia N valores inteiros, imprima o vetor com k elementos por linha, e o maior elemento. O valor de k também deve ser fornecido pelo usuário.
15. Escreva uma função que receba um vetor inteiro v[0..n-1] e os endereços de duas variáveis inteiras, digamos min e max, e deposite nessas variáveis o valor de um elemento mínimo e o valor de um elemento máximo do vetor. Escreva também uma função main que use essa função.
16. Escreva um programa que declare um inteiro, um real e um char, e ponteiros para inteiro, real, e char. Associe as variáveis aos ponteiros (use &). Modifique os valores de cada variável usando os ponteiros. Imprima os valores das variáveis antes e após a modificação.
17. Escreva um programa que declare um vetor de inteiros e um ponteiro para inteiros. Associe o ponteiro ao array. Agora, some mais um (+1) a cada posição do vetor usando o ponteiro (use \*).

18. Escreva um programa que contenha duas variáveis inteiras. Compare seus endereços e exiba o maior endereço.
19. Escreva um programa que contenha duas variáveis inteiras. Leia essas variáveis do teclado. Em seguida, compare seus endereços e exiba o conteúdo do maior endereço.
20. Crie um programa que contenha um array de float contendo 10 elementos. Imprima o endereço de cada posição desse array.
21. Crie um programa que contenha uma matriz de float contendo 3 linhas e 3 colunas. Imprima o endereço de cada posição dessa matriz.
22. 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.
23. 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.
24. 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.
25. Crie uma função que receba dois parâmetros: um vetor e um valor do mesmo tipo do vetor. A função deverá preencher os elementos de vetor com esse valor. Não utilize índices para percorrer o vetor, apenas aritmética de ponteiros.
26. Crie uma função que receba como parâmetro um vetor e o imprima. Não utilize índices para percorrer o vetor, apenas aritmética de ponteiros.
27. 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.