



Lista 06 - Modularização, Ponteiros e Passagem por Referência

1. Fazer uma função *void troca(int *a, int *b)* que recebe dois números inteiros por referência e troca os seus valores. No *main*, leia dois números do usuário, chame a função e mostre os valores trocados.
2. Fazer uma função *void atualizaNota(float *nota)* que recebe uma nota por referência e adiciona 0.5 como bônus, sem ultrapassar o valor 10. No *main*, leia a nota, chame a função e mostre a nota atualizada.
3. Fazer uma função *void calculaOperacoes(int a, int b, int *soma, int *produto)* que recebe dois números inteiros e retorna, por referência, a soma e o produto entre eles. No *main*, leia os dois números, chame a função e mostre os resultados.
4. Fazer uma função *void maiorMenor(int a, int b, int *maior, int *menor)* que recebe dois números inteiros e retorna, por referência, o maior e o menor valor. No *main*, leia os números, chame a função e imprima os resultados.
5. Fazer uma função *void fatorial(int n, int *resultado)* que recebe um número inteiro positivo e retorna, por referência, o seu fatorial. No *main*, leia um número do usuário, chame a função e mostre o fatorial calculado.
6. Fazer uma função *void converteTempo(int segundos, int *horas, int *minutos, int *segRestantes)* que recebe um valor inteiro representando uma quantidade de segundos e converte em horas, minutos e segundos. No *main*, leia um valor em segundos, chame a função e mostre o resultado no formato *hh:mm:ss*.

Continua na próxima página...

Exercícios para entregar

1. Média e Contagem de Números Ímpares

Escreva um programa que leia uma sequência de números inteiros fornecida pelo usuário. A leitura deve ser encerrada quando o usuário digitar o número 0. Ao final, o programa deve exibir:

- A média dos números informados (excluindo o zero);
- A quantidade de números ímpares informados.

2. Pirâmide Numérica Invertida

Faça um programa que leia um número inteiro positivo n e exiba uma pirâmide invertida com n linhas.

Exemplo:

```
Entrada: 4
Saída:
1234
123
12
1
```

3. Verificação de Primos em um Intervalo

Crie um programa que leia dois números inteiros A e B (sendo $A < B$) e exiba todos os números primos existentes nesse intervalo, inclusive os limites se aplicável. Se não houver primos, exiba uma mensagem apropriada.

4. Menu de Operações Matemáticas

Implemente um programa que apresente o seguinte menu de opções e execute as operações até que o usuário escolha a opção 5:

1. Somar dois números
2. Subtrair dois números
3. Multiplicar dois números
4. Dividir dois números
5. Sair

O programa deve solicitar os dois números apenas quando a opção for diferente de 5. Para a divisão, verifique se o divisor é diferente de zero antes de realizar a operação. Em caso de opção inválida, exiba uma mensagem e repita o menu.

5. Quadrado e Cubo de um Número

Escreva uma função:

```
void calculaPotencias(int n, int *quadrado, int *cubo)
```

que recebe um número inteiro e retorna, por referência, o seu quadrado e o seu cubo. No *main*, leia um número, chame a função e exiba os resultados.

6. Contador de Valores Positivos, Negativos e Zeros

Escreva uma função:

```
void classificaNumero(int n, int *positivos, int *negativos)
```

que atualiza, por referência, contadores de números positivos e negativos. No *main*, leia vários números do usuário; o programa deve encerrar a leitura quando o usuário digitar o número 0. Para cada número lido, o *main* deve chamar a função `classificaNumero`. Ao final, exiba as quantidades de números positivos e negativos informados.