GRADUAÇÃO EM ANÁLISE E DESENVOLVIMENTO
PROGRAMAÇÃO DE COMPUTADORES II
Prática 13: Funções

- *Cada programa deve ser efetuado em um arquivo "C" próprio contendo como nome ex1.c para o exercício 1, ex2.c para o exercício 2 e assim por diante;*

1. **Codifique uma função em C chamada *pitagoras()* sem parâmetros e sem retorno. Essa função deve acessar os valores do cateto oposto (variável global – CO) e do cateto adjacente (variável global – CA) calculando o resultado para hipotenusa do triângulo retângulo em questão (armazenando o valor na variável global HIP). O usuário digitará os valores dos catetos e a função Pitágoras será chamada. Por fim, dentro da função *main()*, apresente o valor armazenado na variável HIP. Exemplo:**

   Cateto Oposto: 3
   Cateto Adjacente: 4

   Hipotenusa: 5

2. **Refaça o programa anterior (adaptando a função *pitagoras*) para que ele utilize apenas uma variável global (HIP) que armazenará o valor calculado pela função *pitagoras()*.**

3. **Refaça o programa anterior (adaptando a função *pitagoras*) para que ele não utilize variáveis globais.**

4. **Faça um programa C que possua uma função *fatorial* para calcular o fatorial de um valor inteiro. O valor será digitado pelo usuário na função *main()* e passado a função, que retornará o calculo efetuado. Exemplo:**

   Digite um valor inteiro: 5

   Fatorial de "5": 120

5. **Faça um programa em C que possua uma função *primo* para identificar se um valor é primo ou não. A função deve receber um valor inteiro digitado pelo usuário dentro da função *main()* e retornar *0* caso o valor não seja primo e *1* caso seja primo:**

   Digite um valor inteiro: 13
   Resultado: É PRIMO

   Digite um valor inteiro: 44
   Resultado: NÃO É PRIMO

6. **Faça um programa em C que possua duas funções, *fatorial:* que receberá um valor inteiro como parâmetro e retornará o cálculo do seu fatorial; e *maiorPrimo:* que receberá um valor inteiro como parâmetro e retornará o maior primo encontrado entre 0 e o valor passado como parâmetro. Após isso solicite ao usuário dois valores inteiros e apresente o resultado da multiplicação do fatorial do primeiro pelo maior primo existente zero e o valor do segundo. Exemplo:**

Valor fatorial:	***6***	*(fatorial = 6 x 5 x 4 x 3 x 2x 1 = 720)*
Maior Primo:	***15***	*(maior primo entre 0 e 15 = 13)*
Multiplicação:	**720 x 13 = 9360**	

7. **Faça um programa em C que possua uma função *ordem* para ordenar caracteres. Esta função deverá receber 5 caracteres, converterá todos em caixa alta caso não estejam, e os colocará em ordem alfabética (a impressão deve ser feita dentro da própria função que não possuirá retorno). Exemplo:**

Digite 5 caracteres:	V d K S a
Ordenados:	**A D K S V**

8. ***(DESAFIO)* Faça uma função em C que receba 4 caracteres como parâmetros e retorne um valor inteiro. Ao receber esses caracteres a função deve convertê-los em caixa alta caso eles não estejam, após isso a função deve obter os códigos ASCII de cada caractere e concatená-los em uma variável "*todos*" do tipo inteiro. Após isso a função deve retirar apenas os valores pares encontrados na variável "*todos*" concatenando-os em uma segunda variável "*pares*" que será retornada pela função. Os caracteres devem ser obtidos dentro da função *main()* e repassados a função implementada que retornará o resultado (*pares*) de sua execução a função *main()*. Exemplo:**

Digite 4 Caracteres:	A b C d	*(ASCII: 65 66 67 68 – em caixa alta)* *(Concatenado: 65666768)* *(Apenas os pares: 666668)*
Pares:	***666668***	

9. ***(DESAFIO)* Faça uma função em C que receba dois valores inteiros como parâmetro, obtidos dentro da função *main()*e retorno outro valor inteiro como resultado. A partir desses parâmetros a função deve efetuar sua soma posição por posição, retornando o resultado final (uma variável inteira) calculado. Considere que os números sempre terão o mesmo número de casas decimais e caso a soma de uma das posições ultrapasse 10 a substitua pelo valor 0. Exemplo:**

Primeiro Valor:	***30450***
Segundo Valor:	***43571***
Resultado:	***73901***

10. ***(DESAFIO)*** **Implemente uma função "*recursiva*" que produza com o mesmo resultado de um laço de repetição, como, por exemplo, a função for. A função "*recursiva*" deve receber dois parâmetros inteiros (obtidos do usuário dentro da função *main()*). O primeiro diz respeito ao valor inicial da variável de controle do laço e o segundo se refere a condição de parada dele. Os valores gerados para a variável de controle dentro da função devem ser mostrados através de um *printf()*. Não utilize nenhum laço de repetição em seu código, o conceito a ser implementado é denominado *recursividade*, onde uma função invoca a si própria. Exemplo:**

    Valor Inicial: 0
    Valor Final: 8

    Resultado: 0 – 1 – 2 – 3 – 4 – 5 – 6 – 7