

### Lista de Funções

0) Escreva um algoritmo que lê 2 valores inteiros a e b e usa uma função produto\_notável para devolver ao programa principal o valor desta fórmula  $pn = a^2 + 2.a.b + b^2$ .

1) Escreva um algoritmo que devolve um inteiro que é a soma de 2 valores inteiros maiores que zero. Faça as validações da entrada no programa main.

2) Escreva um algoritmo que devolve um inteiro que é a soma de 3 valores inteiros maiores que zero. Faça as validações da entrada no programa main.

3) Use a função desenvolvida na questão 1 para calcular Fibonacci. No programa principal:

- o usuário passa um valor N para ser o número de termos de Fibonacci que deseja visualizar
- os valores iniciais são fixados ( 1 e 1 ).

Conte quantas vezes a função é chamada para informar o número de termos calculados (atender o N solicitado do usuário).

4) C

Desenvolva o exercício da Conjectura de Collatz usando as funções parCollatz ( divide x por 2 e retorna o valor desta divisão) e imparCollatz ( que lê um int x e devolve como retorno o valor de  $3*x + 1$ ). A entrada para cada função é um inteiro x e a saída , outro inteiro ( resultado da operação adequada).

*Cabeçalho das funções :*

*int parCollatz( int x );*

*int imparCollatz(int x);*

Chame essas funções para resolver o problema original abaixo:

#### **Conjectura de Collatz ou Problema do $3n + 1$**

Leia um N inteiro entre 1 e 1 milhão Esse número deve ser dividido por 2 se for par , senão deve ser aplicado na fórmula  $3*N + 1$  Deve-se parar quando chegar em 1 .

Informe qual o maior valor obtido nos fatores calculados e quantos passos foram feitos até chegar no valor 1 Informe quantos números calculados são ímpares e quantos são pares.

Exemplo1.....

( entrada) N=12

12, 6, 3, 10, 5, 16, 8, 4, 2, 1

**(saída)** Há 10 passos e o maior valor é 16.

Exemplo2.....

( entrada) n=27

27, 82, **41**, 124, 62, **31**, 94, **47**, 142, **71**, 214, **107**, 322, **161**, 484, 242, **121**, 364, 182, **91**, 274, **137**, 412, 206, **103**, 310, **155**, 466, **233**, 700, 350, **175**, 526, **263**, 790, **395**, 1186, **593**,

1780, 890, **445**, 1336, 668, 334, **167**, 502, **251**, 754, **377**, 1132, 566, **283**, 850, **425**, 1276, 638, **319**, 958, **479**, 1438, **719**, 2158, **1079**, 3238, **1619**, 4858, **2429**, 7288, 3644, 1822,

**911**, 2734, **1367**, 4102, **2051**, 6154, **3077**, **9232**, 4616, 2308, 1154, **577**, 1732, 866, **433**, 1300, 650, **325**, 976, 488, 244, 122, **61**, 184, 92, 46, **23**, 70, **35**, 106, **53**, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, **1**.

**( saída)** Há 111 passos e o maior valor é 9232

5) Escreva uma função recebe como parâmetro de entrada um número int x e retorna se ele é primo ou não. Retorne 0 se for primo, 1 para outro caso.

*Cabeçalho da função*

*int eh\_primo( int x);*

Teste o código no programa principal abaixo:

```
int main( ){
    int N;
    printf("\nDigite um valor inteiro maior que 1 :\n");
    if ( N > 1 ){
        scanf("%i", &N );
        if ( eh_primo(N) == 0)
            printf("\n %i primo", N);
        else
            printf("\n %i não primo", N);
    }else
        printf("\n 1 não primo");
    return 0;
}
```

6) Escreva uma função que recebe dois valores inteiros como parâmetros de entrada ( a e b ) e testa se o valor de b é um divisor do valor a . Se a for divisor de b, a função deve retornar 0, senão 1.

*Cabeçalho da função:*

*int divisor( int a , int b);*

Teste o código no programa principal abaixo:

```
int main( ){
    int a, b, resposta;
    printf("\nDigite 2 valores inteiros e maiores que zero:\n");
    scanf("%i %i", &a, &b );
    if ( a >= b )
        resposta = divisor( a , b )
    else
        resposta= divisor ( b, a );
    if ( resposta ==0 )
        printf("\ Divide");
    else
        printf("\ Nao Divide");
    return 0;
}
```

7) Escreva uma função que lê um inteiro N como parâmetro de entrada e devolve o próximo valor primo após esse N.

*Cabeçalho da função:*

*int proximoPrimo ( int N ) ;*

Entrada	Saída
1	2
2	3
3	5
4	5
6	7
7	11

8) Escreva um algoritmo que usará uma função para encontrar o primeiro valor de MMC( Mínimo Multiplo Comum ) de 3 valores inteiros a, b e c - fornecidos pelo usuário. Use a regra da fatoração pelos primos.

Dica

*int MMC ( int a , int b , int c )*

- MMC ler 3 valores inteiros e dividir por primos ( fatorar) ;
- cada fator divisor deve ser multiplicado num acumulador mmc;
- Entradas: 3 valores inteiros a , b, c
- Saída: primeiro MMC encontrado das entradas a, b, c
- Veja a 2a forma de MMC para usar em <https://mundoeducacao.uol.com.br/matematica/regra->

<pratica-para-calcular-mmc.htm>

- use a função proximoPrimo da questão 7 para gerar os divisores

Entrada	Saída
12 18 24	M.M.C. (12, 18, 24) = $2 \times 2 \times 2 \times 3 \times 3 = 72$
15 25 70	M.M.C. (15, 25, 70) = $2 \times 3 \times 5 \times 5 \times 7 = 1\ 050$