

UNIVERSIDADE FEDERAL DO MARANHÃO

FUNÇÃO

1. Crie uma função que recebe como parâmetro um número inteiro e devolve o seu dobro.
2. Faça uma função que receba 3 números inteiros como parâmetro, representando horas, minutos e segundos, e os converta em segundos.
3. Faça uma função que receba dois valores numéricos e um símbolo. Este símbolo representará a operação que se deseja efetuar com os números. Se o símbolo for + deverá ser realizada uma adição, se for - uma subtração, se for / uma divisão e se for * será efetuada uma multiplicação.
4. Crie uma função que receba como parâmetro um valor inteiro e gere como saída n linhas com pontos de exclamação, conforme o exemplo abaixo (para $n = 5$):

```
!  
!!  
!!!  
!!!!  
!!!!!
```

5. Escreva uma função que gera um triângulo de altura e lados n e base $2*n-1$. Por exemplo, a saída para $n = 6$ seria:

```
  *  
 ***  
*****  
*****  
*****  
*****  
*****
```

6. Faça uma função chamada 'simplifica' que recebe como parâmetro o numerador e o denominador de uma fração. Esta função deve simplificar a fração recebida dividindo o numerador e o denominador pelo maior fator possível. Por exemplo, a fração 36/60 simplifica para 3/5 dividindo o numerador e o denominador por 12. A função deve modificar as variáveis passadas como parâmetro.

7. Faça uma função que receba um número inteiro positivo n e retorne o fatorial exponencial desse número. Um fatorial exponencial é um inteiro positivo n elevado a potência de $n - 1$, que por sua vez é elevado à potência de $n - 2$ e assim em diante. Ou seja: $n^{(n-1)(n-2) \dots}$

8. Faça uma função que recebe, por parâmetro, uma matriz $A[3][3]$ e retorna a soma dos elementos da sua diagonal principal e da sua diagonal secundária.

9. Considerando a estrutura:

```
struct Ponto{  
    int x;  
    int y;  
};
```

Para representar um ponto em uma grade 2D, implemente uma função que indique se um ponto p está localizado dentro ou fora de um retângulo. O retângulo é definido por seus vértices inferior esquerdo $v1$ e superior direito $v2$. A função deve retornar 1 caso o ponto esteja localizado dentro do retângulo e 0 caso contrário. Essa função deve obedecer ao protótipo:

```
int dentroRet (struct Ponto* v1, struct Ponto* v2, struct Ponto* p);
```

10. Considerando a estrutura:

```
struct Vetor{  
    float x;  
    float y;  
    float z;  
};
```

Para representar um vetor no R^3 , implemente uma função que calcule a soma de dois vetores. Essa função deve obedecer ao protótipo:

```
void soma (struct Vetor* v1, struct Vetor* v2, struct Vetor* res);
```

onde os parâmetros $v1$ e $v2$ são ponteiros para os vetores a serem somados, e o parâmetro res é um ponteiro para uma estrutura vetor onde o resultado da operação deve ser armazenado.