

Lista de Exercícios – Função e Recursividade

Monitor: Rafael Sian de Freitas

- 1** – Faça uma função que receba como parâmetro um número inteiro e imprima esse número.
- 2** – Faça uma função que receba como parâmetro um número inteiro e retorne o seu quadrado.
- 3** – Faça uma função que verifica se um número recebido é primo ou não.
- 4** – Faça uma função que receba 3 valores float e retorne a soma dos dois primeiros multiplicado pelo terceiro.
- 5** – Faça uma função retorne o fatorial de um número recebido.
- 6** – Faça uma função que verifica se um número é perfeito.
- 7** – Faça uma função que verifica se um número é triangular.
- 8** – Faça uma função similar ao do exercício 5, que agora verifica quantos números primos existem de 1 até N. O número N tem que ser passado para a função.
- 9** – Faça uma função que receba dois números e retorne o maior deles.
- 10** – Faça uma função que receba três números e retorne o menor deles.
- 11** – Faça uma função que informa se um número é positivo ou negativo.
- 12** – Faça uma função que receba um número inteiro positivo N e retorne a soma de todos os números até N.
Exemplo:

N = 4.
Retorno da função: 1+2+3+4
- 13** – Faça uma função que receba um número inteiro e retorne quantos algarismos esse número possui.
- 14** – Faça uma função que retorne a quantidade de minutos que existem entre dois horários informados pelo usuário.
Exemplo:

Horário 1: 15:20
Horário 2: 16:30
Retorno: 70 minutos
- 15** – Faça uma função que receba um vetor de tamanho 10 e retorne o maior valor que existe dentro do vetor.

Exercícios Resolvidos

1 – Faça uma função que receba como parâmetro dois números e retorne a soma desses dois números.

```
int somaNumero (int num1, int num2)
{
    int soma;
    soma = num1 + num 2;
    return soma;
}
```

2 – Faça uma função que sorteia um número qualquer.

```
int sorteiaNumero ()
{
    num = srand();
    return num;
}
```

3 – Faça uma função que verifica se um número recebido é par ou não.

```
void paridade (int num)
{
    if (num % 2 == 0)
    {
        printf ("O numero %d e par!", num);
    }
    else
    {
        printf ("O numero %d nao e par!", num);
    }
}
```

Desafios

1 – Faça uma função recursiva que receba a posição de um número para se calcular na sequência de Fibonacci.

Exemplo:

N = 3

Saída: 2