

Lista de Exercícios: Recursividade

1. Qual o resultado de **tail(7)**?

```
void tail(int i)
{
    if(i>0){
        cout << i << " ";
        tail(i-1);
    }
}
```

2. Escreva uma função recursiva que imprima números inteiros de 1 a 100.
3. Escreva uma função recursiva para elevar qualquer número x a uma potência inteira não negativa n . A definição mais natural dessa função é dada por:

$$x^n = \begin{cases} 1 & \text{se } n=0 \\ x \cdot x^{n-1} & \text{se } n>0 \end{cases}$$

4. Escreva uma função recursiva $\text{MDC}(n, m)$ que retorne o maior divisor comum de dois inteiros n e m , de acordo com as seguintes definições:

$$\text{MDC}(n,m) = \begin{cases} m & \text{se } m \leq n \text{ e } n \bmod m = 0 \\ \text{MDC}(m,n) & \text{se } n < m \\ \text{MDC}(m, n \bmod m) & \text{caso contrário} \end{cases}$$

5. Escreva uma função recursiva que imprima a tabuada de um número N lido.
6. Determine o que faz a função recursiva a seguir:

```
int recursive(int n)
{
    if(n<=0) return 1;
    else return recursive(n-1)+recursive(n-1);
}
```

7. Faça uma versão recursiva da seguinte função:

```
void imprime(int v[], int tamanho)
{
    int i;
    for(i=0; i<tamanho; i++)
        cout << v[i] << " ";
}
```

8. Defina uma função de busca simples recursiva em vetor.
9. Supondo que os elementos de um vetor estão ordenados de forma crescente, defina uma função de busca recursiva nesse vetor.
10. Escreva uma função recursiva para converter um número decimal para a forma binária. Uma maneira simples de resolver o problema é dividir o número decimal sucessivamente por 2 e pegar o resto da i -ésima divisão, da direita para a esquerda. Por exemplo, para o número 12 temos: $12/2=6$, resto 0; $6/2=3$, resto 0; $3/2=1$, resto 1; $1/2=0$, resto 1. Portanto, o número 12 em binário é 1100.