

Programação Estruturada

- Encontro 11 -

Sistemas de Informação
Prof.º Philippe Leal
philippeleal@yahoo.com.br



Definição

- Chamamos de **recursividade** ou **recursão** quando uma função chama a si mesma.
- Em uma função recursiva, a cada chamada é criada na memória uma nova ocorrência da função com comandos e variáveis “isolados” das ocorrências anteriores.
- A função é executada até que todas as ocorrências tenham sido resolvidas.



Definição

- Grande parte das funções recursivas são compostas por um **caso base** e pelas **chamadas recursivas**:
 - **Caso base** (ou **condição de parada**): é o caso mais simples. É usada uma condição em que se resolve o problema com facilidade, onde o resultado é imediatamente conhecido;
 - **Chamadas recursivas**: procuram resolver um subproblema do problema original de tal forma que convergem para o caso base.



Vantagens e Desvantagens

- **Vantagens da Recursividade**

- Torna a escrita do código mais simples e elegante, tornando-o fácil de entender e de manter.

- **Desvantagens da Recursividade**

- Quando o loop recursivo é muito grande, é consumida muita memória nas chamadas a diversos níveis de recursão, pois cada chamada recursiva aloca memória para os parâmetros, variáveis locais e de controle;

- Em muitos casos uma solução iterativa gasta menos memória e torna-se mais eficiente em termos de performance do que usar recursão.



Exemplo

- Faça uma função que calcule e retorne a soma de 1 até n , sendo $n \geq 1$.



Exemplo

- Faça uma função que calcule e retorne a soma de 1 até n , sendo $n \geq 1$.

➤ **Função iterativa:**



Exemplo

- Faça uma função que calcule e retorne a soma de 1 até n , sendo $n \geq 1$.

➤ **Função iterativa:**

```
int calculaSoma(int n){  
  
    int i, soma = 0;  
    for(i = 1; i <= n; i++)  
        soma += i;  
  
    return soma;  
}
```



Exemplo

- Faça uma função que calcule e retorne a soma de 1 até n , sendo $n \geq 1$.

➤ **Função recursiva:**



Exemplo

- Faça uma função que calcule e retorne a soma de 1 até n , sendo $n \geq 1$.

➤ **Função recursiva:**

```
int calculaSoma(int n){  
  
    if(n == 1)  
        return 1;  
  
    else  
        return n + calculaSoma(n-1);  
  
}
```



Exemplo

- Faça uma função que calcule e retorne a soma de 1 até n , sendo $n \geq 1$.

➤ **Função recursiva:**

```
int calculaSoma(int n){
```

```
    if(n == 1)  
        return 1;
```

Caso Base

```
else
```

```
    return n + calculaSoma(n-1);
```

Chamada Recursiva

```
}
```



Exercícios

- 1) Faça um algoritmo recursivo para calcular o fatorial de n , onde $n \geq 0$.
- 2) Dado um número inteiro n ($n > 0$), faça um algoritmo recursivo que calcule a soma dos primeiros n cubos, ou seja, que calcule $S(n)$, onde:

$$S(n) = \sum_{i=1}^n i^3 = 1^3 + 2^3 + 3^3 + \dots + n^3$$