

Recursividade

 **adrierj@gmail.com** (não compartilhado) [Alternar conta](#)



***Obrigatório**

Exemplo

Recursividade é a possibilidade de uma função chamar a si mesma.

Observe o código da imagem:

Fatorial de um número

Exemplo 1: Fatorial

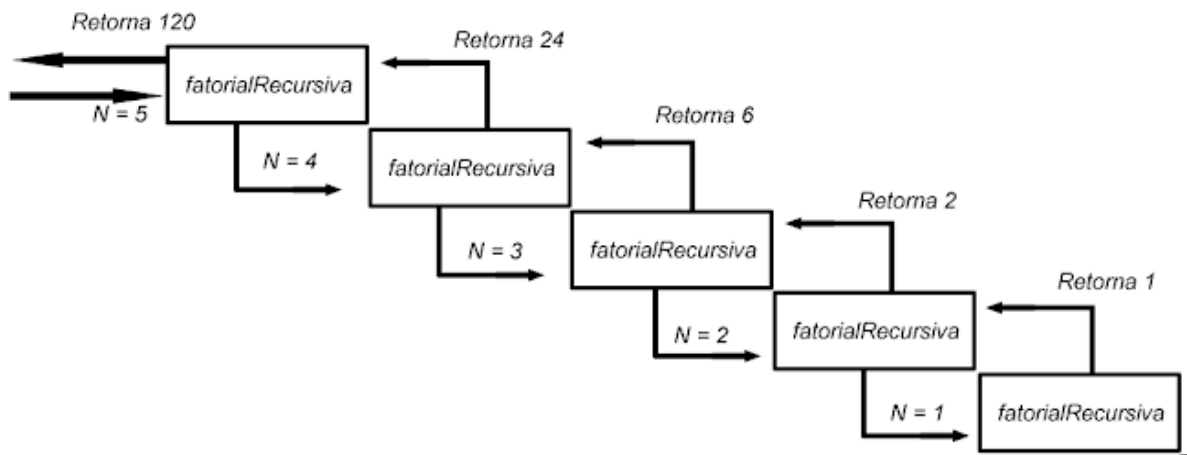
```
/* calculo do fatorial – função recursiva */  
int fat(int n)  
{  
  if (n==1){  
    return 1;  
  }else{  
    return n*fat(n-1);  
  }  
}
```



Explicando

Como apresentado na imagem, a função `fat` chama a si mesma na linha `return n*fat(n-1)`. Ou seja, considerando esse trecho, o programa irá entrar na função várias vezes. O ponto de parada, ou caso base para esse exemplo é o `if(n==1)`. Ao entrar nessa condicional a função para de chamar a si mesma e o valor um é retornado para a chamada anterior. Acontece o que a próxima figura ilustra:

Considerando a entrada 5 para `n` na função acima



Explicando a ilustração

Note que, tomando como valor para `n` o número 5, a função será sempre chamada, até que `N` seja 1. Nesse ponto acontece o que foi explicado anteriormente, o retorno será 1, e as demais chamadas também irão devolver seus valores de retorno. Acontece nessa execução o que chamamos de pilha de execução (a primeira chamada será a última a oferecer o retorno).



Exemplo: Chamada: `main()` -> `funcao_1()` -> `funcao_2()` -> `funcao_3()` Retorno:
`funcao_3()` -> `funcao_2()` -> `funcao_1()` -> `main()`

Exemplo: A Pilha de Execução de um Programa

```
main() {  
    funcao_1();  
}  
  
funcao_1() {  
    funcao_2();  
}  
  
funcao_2() {  
    funcao_3();  
}  
  
funcao_3() {  
    ...  
} ←
```

Pilha de Execução
funcao_2
funcao_1
main



Vídeo

[Programação Estruturada na Linguagem C (Parte 11)] Recur...



código do vídeo

```
#include <stdio.h>
```

```
int fat(int n){
```

```
/*
```

```
n=5 -> n*fat(n-1) = 5*24 = 120
```

```
n=4 -> n*fat(n-1) = 4*6 = 24
```

```
n=3 -> n*fat(n-1) = 3*2 = 6
```

```
n=2 -> n*fat(n-1) = 2*1 = 2
```

```
n=1 -> 1
```

```
*/
```

```
if (n==1){
```

```
    return 1;
```

```
}
```

```
return n*fat(n-1);
```

```
}
```

```
main( )
```

```
{
```

```
printf("Fatorial\n");
```

```
int n;
```

```
scanf("%d", &n);
```

```
// n=5; 5! -> 5x4x3x2x1 = 120
```

```
printf("O valor da soma total eh %d", fat(n));
```

```
return(0);
```

```
}
```

Por definição, o fatorial de 0 é 1 ($0!=1$). Altere a função do programa apresentado aqui no material e no vídeo para que essa definição seja atendida pelo programa. Ou seja, se a entrada para n for 0, qual o retorno? *

Sua resposta

Voltar

Enviar

Limpar formulário

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. [Denunciar abuso](#) - [Termos de Serviço](#) - [Política de Privacidade](#)



Google Formulários

