

FUNÇÕES RECURSIVAS EM C

Recursividade é quando uma função chama a si mesma. É uma técnica que divide o problema em versões menores. Ela é utilizada sempre que for possível expressar a solução de um problema em função do próprio problema.

Sendo assim, ela funciona da seguinte forma: a cada chamada é construída na memória uma nova ocorrência da função com comandos e variáveis “isolados” das ocorrências anteriores. A função só termina quando todas as ocorrências tenham sido resolvidas.

Fazer conseguir pará-lo, é um dos problemas da recursividade. Um programador inexperiente pode cair facilmente em um loop infinito.

A recursividade é composta por:

- **Caso base:** caso mais simples, usando uma condição em que se resolve o problema com facilidade.
- **Chamadas Recursivas:** procuram simplificar o problema de tal forma que convergem para o caso base.

A vantagem do uso da recursividade é que escrita do código mais simples e elegante, sendo mais fácil entendê-lo.

Porém, algumas desvantagens é, como já dito antes, possível que ocorra um loop recursivo, e isso faz com que consuma muita memória nas chamadas das funções. Além disso, em muitos casos, uma solução iterativa gasta menos memória, sendo de melhor performance do que a recursão.

Exemplo de uma função que calcula fatorial e usa recursividade em C:

```
int fatrec(int x){  
    if (x == 0) {  
        return 1;  
    }  
    else {  
        int f;  
        f = x * fatrec(x-1);  
        return f;  
    }  
}
```

```
main () {
```

```
int n5, f;  
printf("Numero: ");  
scanf("%d",&n5);  
f = fatrec(n5);  
printf("fat(%d) = %d",n5,f);  
}
```

BIBLIOGRAFIA

<https://www.codingame.com/playgrounds/24988/programacao-c/funcoes-recursivas>
<http://linguagemc.com.br/recurividade-em-c/>