Universidade de São Paulo – São Carlos
Instituto de Ciências Matemáticas e de Computação

Recursão em C

Vânia de Oliveira Neves

Slides baseados no material gentilmente cedido pela Profa.
Simone Senger Souza

Recursão em C

uma função é dita recursiva quando dentro do seu código existe uma chamada para si mesma

Exemplo:

F = [C, F] → recursão direta

F1 = [C1, F2]
F2 = [C2, F3]
F3 = [C3, F4]
…
Fn = [Cn, F1] → recursão indireta

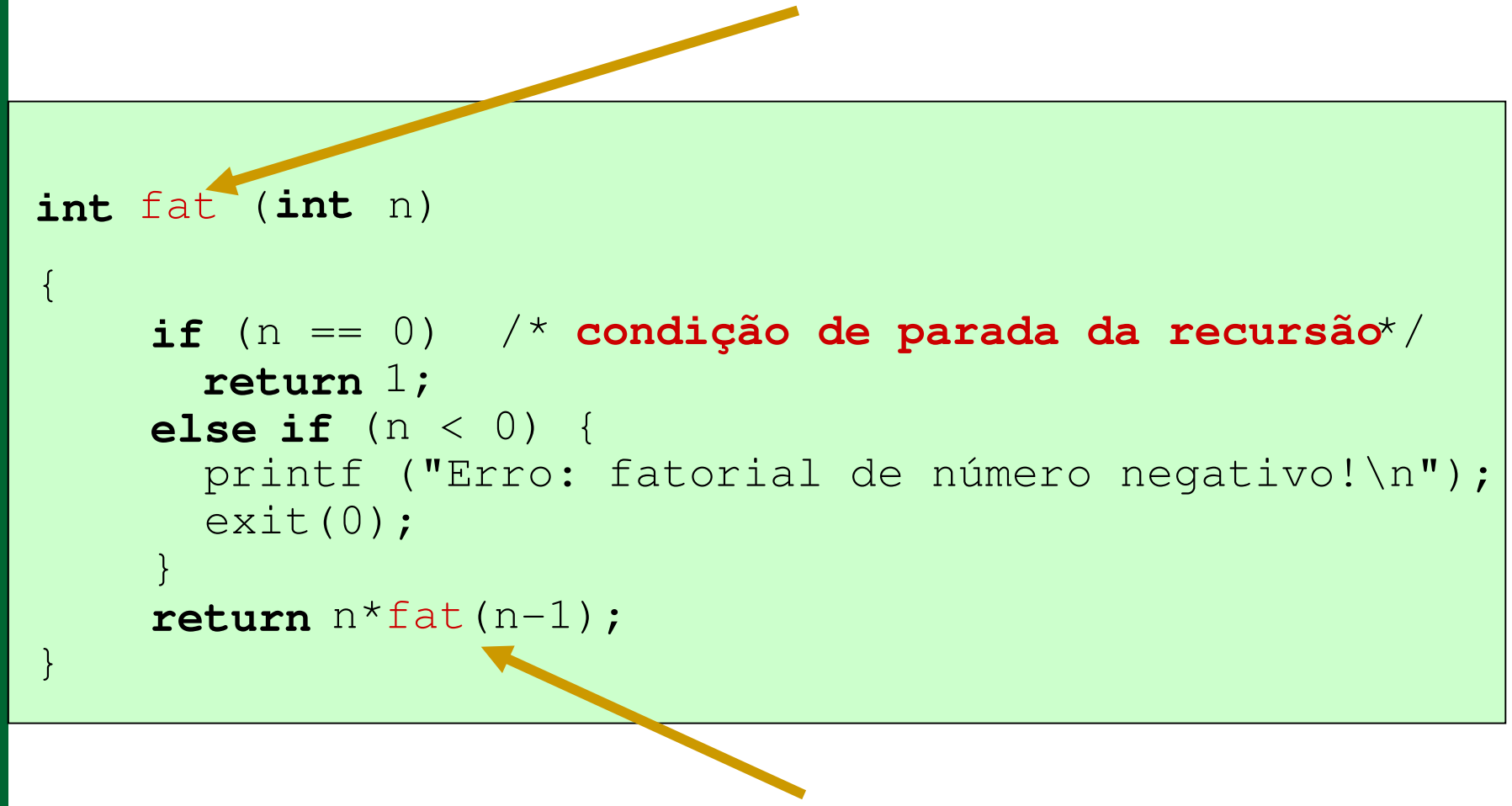
Recursão em C

- A recursão é uma técnica que define um problema em termos de uma ou mais versões menores deste mesmo problema
- Esta ferramenta pode ser utilizada sempre que for possível expressar a solução de um problema em função do próprio problema

Recursão em C

- Ao definir uma função recursiva, dividimos o problema da seguinte maneira:
  - **solução trivial**: dada por definição, não necessita de recursão para ser obtida. Esta parte do problema é resolvida pelo conjunto de comandos.
  - **solução geral**: parte do problema que em essência é igual ao problema original, sendo porém menor. A solução pode ser obtida por uma chamada recursiva R(x-1).
- Ex: fatorial
  - solução trivial: 0! = 1 → implica na condição de parada da recursão!
  - solução geral: n! = n*(n-1)!

Exemplo Fatorial



```
int fat (int n)

{
     if (n == 0)   /* condição de parada da recursão*/
       return 1;
     else if (n < 0) {
       printf ("Erro: fatorial de número negativo!\n");
       exit(0);
     }
     return n*fat(n-1);
}
```

Analisando recursividade

Vantagens X Desvantagens

- Um programa recursivo é mais elegante e menor que a sua versão iterativa, além de exibir com maior clareza o processo utilizado, desde que o problema ou os dados sejam naturalmente definidos através de recorrência.
- Por outro lado, um programa recursivo exige mais espaço de memória e é, na grande maioria dos casos, mais lento do que a versão iterativa.