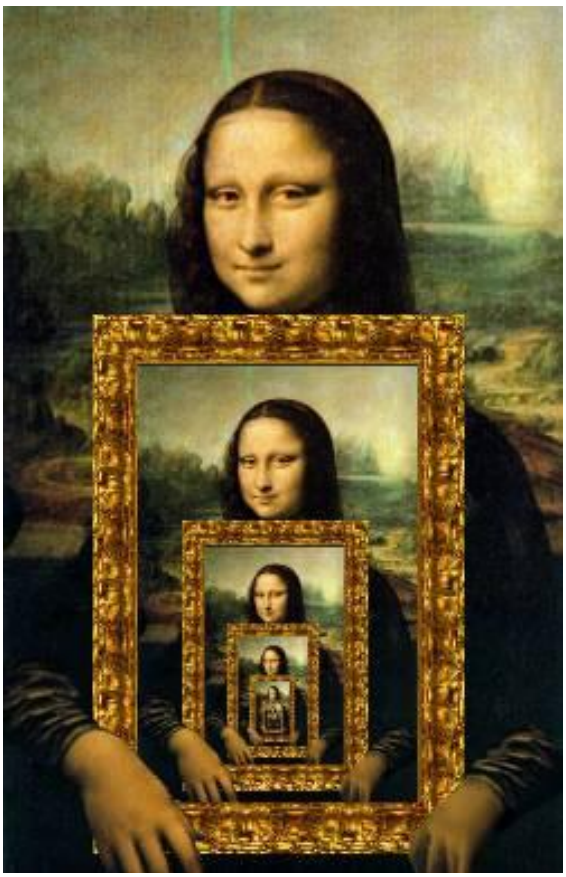
RECURSIVIDADE

Um objeto é dito recursivo se ele consistir parcialmente ou for definido em termos de si próprio. A recursão é encontrada não apenas na matemática, mas também no dia-a-dia. Quem nunca viu uma figura que contenha ela mesma?



Portanto, um método recursivo é um método para ser aplicado a uma estrutura, envolve a aplicação dele mesmo às suas estruturas componentes. Em termos de programação, isto implica em procedimentos/funções que chamam a si mesmos.

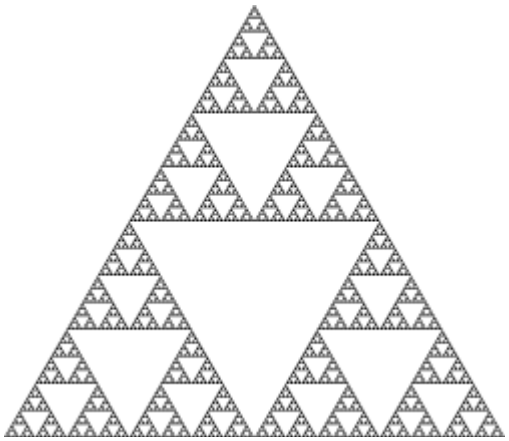
Recursão é uma técnica particularmente poderosa em definições matemáticas. Alguns exemplos são os números naturais, estrutura em árvores e algumas funções (exemplo: cálculo de fatorial).

1) Números naturais

2) Cálculo do fatorial

3) Árvores binárias

4) Triângulo de Sierpinski: uma recursão fechada de triângulos formando uma reticulada geométrica.

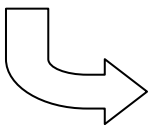


O poder da recursão deve-se evidentemente, à possibilidade de se definir um conjunto finito de objetos através de uma formulação finita.

Algoritmos recursivos são, no entanto, especialmente adequados quando o problema a ser resolvido, a função a ser calculada ou a estrutura de dados a ser processada estejam definidos de forma recursiva.

Toda vez que uma rotina é ativada recursivamente, um novo conjunto de variáveis locais (ou existentes) é criado. Ainda que estas variáveis tenham os mesmos valores que os elementos correspondentes, no conjunto local a uma instância prévia da rotina, seus valores são diferentes e o conflito de nomes, por ventura existente, é evitado pelas aplicações das regras de escopo de identificadores: os identificadores sempre referem ao conjunto de objetos mais recentemente criados. A mesma regra vale para os parâmetros passados por valor das rotinas.

Um requisito fundamental é que chamadas recursivas de uma rotina sejam sujeitas a uma condição, que em determinado momento se torne FALSA.


CRITÉRIO DE PARADA

Em condições práticas, é vital mostrar que o nível mais profundo da recursão seja não apenas finito, mas também relativamente pequeno (memória para armazenar variáveis, estado corrente do processamento).

Importante: Quando NÃO usar recursão.

Algoritmos recursivos são particularmente apropriados quando o problema a que se destina for definido em termos recursivos. Isto não significa, no entanto, que tais definições recursivas garantam ser o algoritmo a maneira mais eficiente de resolver o problema.

Dessa forma, deve-se evitar o uso da recursão sempre que for conhecida uma solução óbvia que utilize a técnica de iteração. Entretanto, há muitas boas aplicações para recursão, como os algoritmos exaustivos, o problema das 8 rainhas.

Exemplos:

1) Cálculo do somatório

$$\sum_{i=1}^{n} i = 1 + 2 + 3 + \cdots + n$$

Modo Iterativo	Modo Recursivo

2) Cálculo do Fatorial

Modo Iterativo	Modo Recursivo

3) Cálculo do m.d.c. (máximo divisor comum)

Modo Iterativo	Modo Recursivo

4) Cálculo de x^y, $x \in R$ e $y \in N$.

Modo Iterativo	Modo Recursivo

5) Cálculo da soma dos elementos de um vetor

Modo Iterativo	Modo Recursivo

PROTÓTIPOS OU CABEÇALHOS DE ROTINAS

Um protótipo faz uma rotina tornar-se conhecida sem realmente especificar o seu corpo (parte executável).

A partir da declaração do protótipo, outras rotinas podem chamar a rotina declarada como protótipo, tornando possível uma recursão mútua.

A declaração do protótipo da rotina pode omitir o nome dos parâmetros, mas não seus tipos.

Exemplo:

```
void flip(int); // protótipo da rotina

 void flop(int n) {
   printf("Flop\n");
   if (n > 0) flip(n - 1);
 }

void flip(int n) {
   printf("Flip\n");
   if (n > 0) flop(n - 1);
 }
```

EXERCÍCIOS

1 Escreva uma função recursiva que imprima os n primeiros números naturais na tela.

2 Escreva uma função recursiva que imprima os n primeiros números pares na tela.

3 Fibonacci (matemático da Renascença italiana) estabeleceu uma série curiosa de números para modelar o número de casais de coelhos em sucessivas gerações. Em cada geração, o número pode ser obtido através dos das 2 gerações anteriores. Assumindo que nas primeiras duas gerações só existe um casal de coelhos, os sucessivos números de Fibonacci são 1,1,2,3,5,8,13,21,34,55,89, ... Estes números ocorrem em vários processos biológicos (e não só), por exemplo, no número de pétalas de algumas flores.

Escreva um algoritmo iterativo e um recursivo que retorne o n=-ésimo número da sequencia de Fibonacci.

4 Reescreva a função abaixo tornando-a recursiva:

```
int Digitos (int N){
    int cont = 1;
   while (abs(N) > 9) {
        N = N / 10;
        cont++;
    }
    return cont;
}
```

5 A Torre de Hanói é um jogo com uma base histórica citada num ritual praticado por sacerdotes brâmanes para predizer o fim do mundo. O jogo inicia-se com uma série de anéis de ouro de tamanhos decrescentes, empilhados numa haste presa numa tábua (os sacerdotes brâmanes usavam 64 anéis). O objetivo é empilhar todos os anéis numa segunda haste em ordem decrescente de tamanho. Antes que isso possa ser feito, o fim do mundo chegará. Uma terceira haste está disponível para uso como armazenamento intermediário. Construa um programa recursivo para o problema da torre de Hanói. O problema consiste em transferir N discos, utilizando três hastes, que estão na haste A para a haste C, utilizando as seguintes regras:

- pode-se mover somente um disco de cada vez;
- nenhum disco deve ser colocado sobre um disco menor;
- qualquer disco pode ser movido de qualquer haste para qualquer haste, respeitando a regra acima.