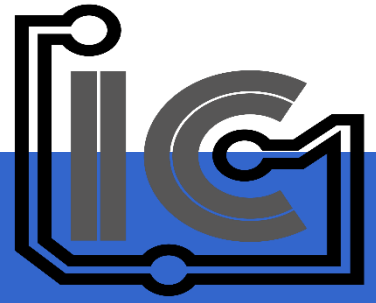
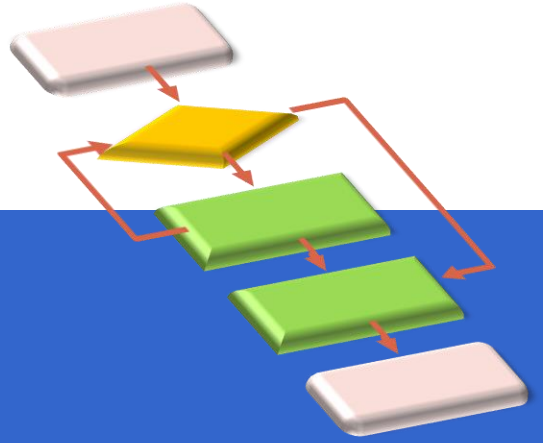


INSTITUTO DE COMPUTAÇÃO



Algoritmos II

RECURSIVIDADE
Implementação

Prof.ª Vanessa de Oliveira Campos

Subprograma Recursivo

- O cálculo do fatorial de um número *n*, inteiro positivo, é realizado por sucessivas multiplicações:

n! = n * (n-1) * (n-2) * ... * 3 * 2 * 1

Fatorial

```
Funcao fatorial(n: inteiro): inteiro
var
      num, val_fat : inteiro
inicio
      val_fat ← 1
      para num de n até 1 faca
            val_fat ← val_fat * num
      fimpara
      retorne val_fat
fimfuncao
```

Subprograma Recursivo

$4! = ?$

$4! = 4 * 3 * 2 * 1 \rightarrow 4! = 4 * 3!$

$3! = 3 * 2 * 1 \rightarrow 3! = 3 * 2!$

$2! = 2 * 1 \rightarrow 2! = 2 * 1!$

$1! = 1 * 0!$

$0! = 1$

Subprograma Recursivo

- Definição recursiva de fatorial:

  - se n = 0 então

    fatorial(n) = 1

  - se n > 0 então

    fatorial(n) = n * fatorial(n-1)

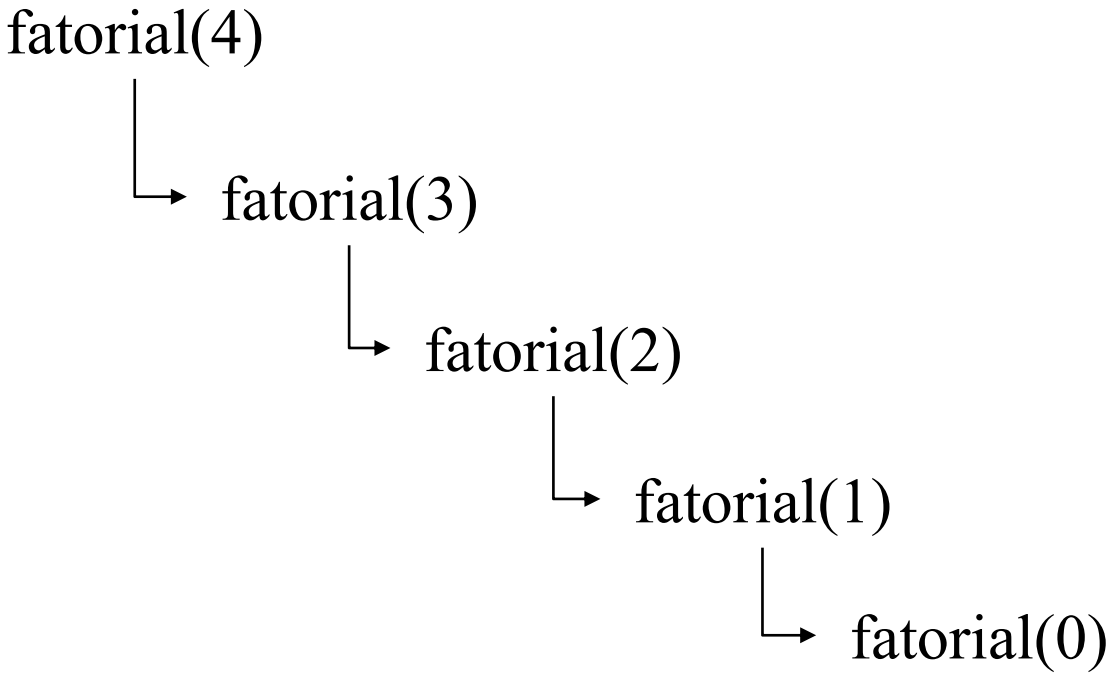
Subprograma Recursivo

```
Funcao fatorial(n: inteiro): inteiro
inicio
      se n = 0 então
            retorne 1
      senão
            retorne n * fatorial(n-1)
      fimse
fimfuncao
```

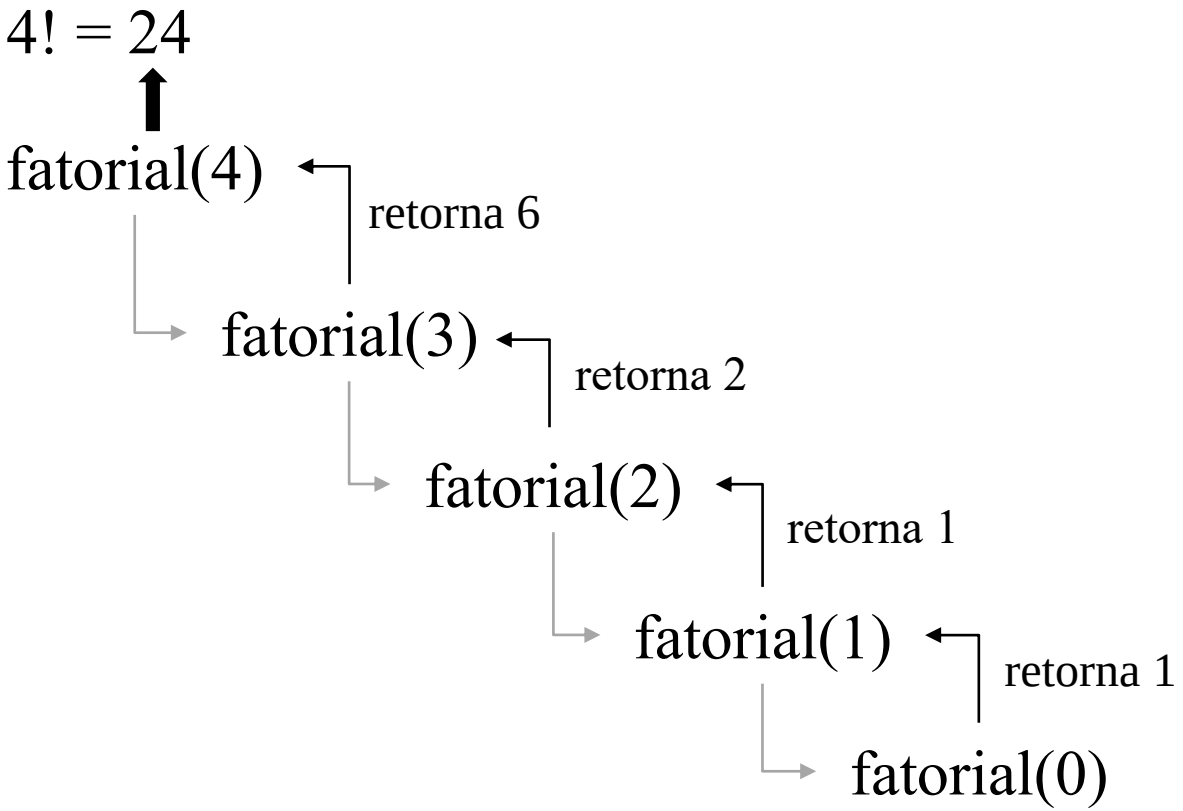
chamada recursiva

Subprograma Recursivo

4! = ?



Subprograma Recursivo



Parada de chamadas recursivas

Atenção!

Exemplo:

```
Funcao Z(n: inteiro): inteiro
inicio
   retorne Z(n + 1) { CHAMADA RECURSIVA }
fimfuncao
```

Subprogramas recursivos sempre devem ter uma **condição de parada**!

Sequência de Fibonacci

- A sequência

[0, 1, 1, 2, 3, 5, 8, 13, 21, ...]

é conhecida como *série de Fibonacci* e tem aplicações teóricas e práticas (alguns padrões na natureza parecem segui-la).

Sequência de Fibonacci

- Definição recursiva:

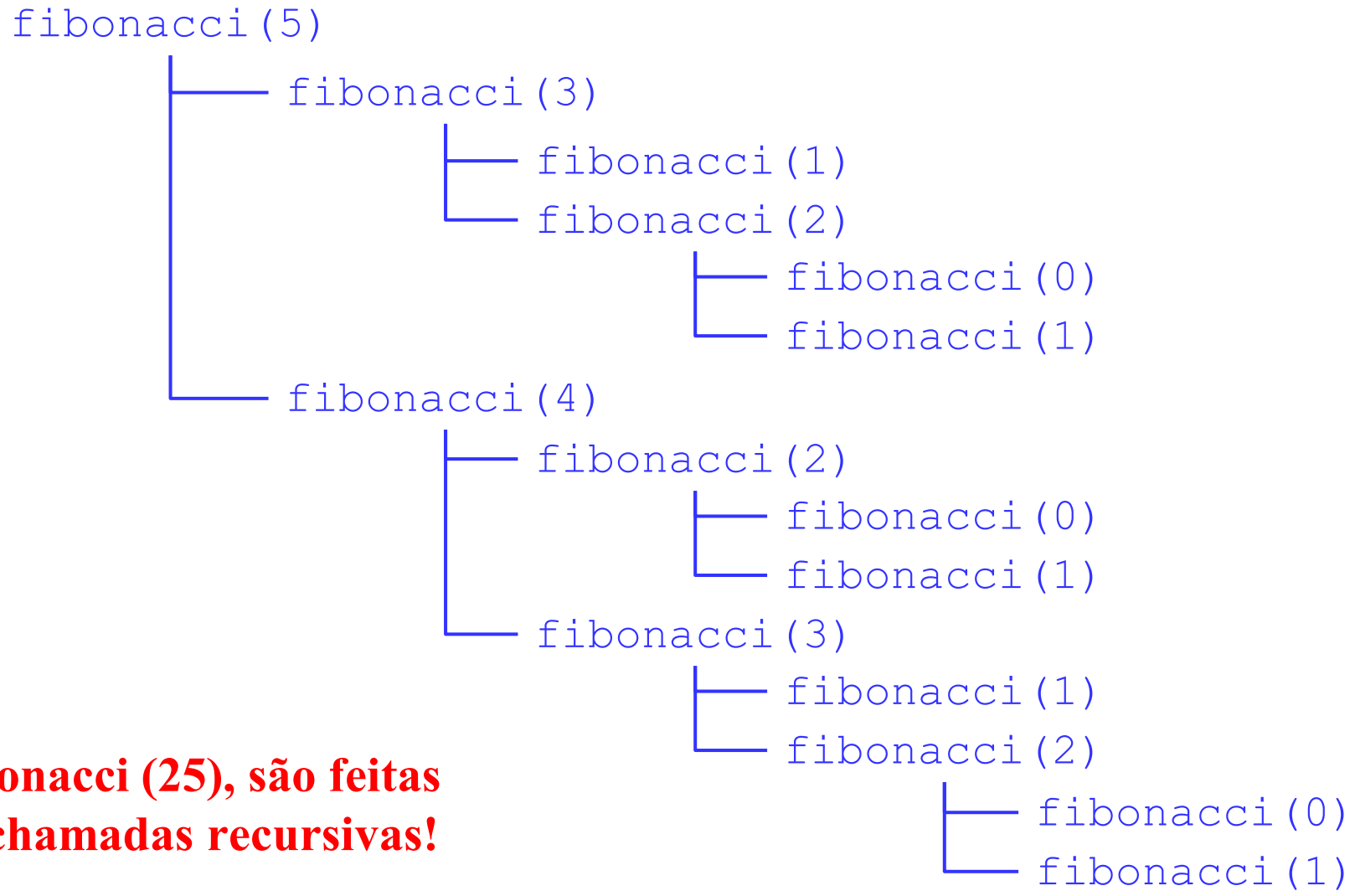
$$Fib(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ Fib(n-1) + Fib(n-2) & \text{se } n > 1 \end{cases},$$

Sequência de Fibonacci

```
funcao fibonacci(n: inteiro): inteiro
inicio
      se n = 0 ou n = 1 então
            retorne n
      senao
            retorne fibonacci(n-2) + fibonacci(n-1)
      fimse
fimfuncao
```

Sequência de Fibonacci

Diagrama de execução de fibonacci(5):



Para fibonacci (25), são feitas 242784 chamadas recursivas!

Sequência de Fibonacci


```
funcao fibonacci(n: inteiro): inteiro
var
      i, F1, F2, F : inteiro
início
      se n = 0 ou n = 1 então
            retorne 1
      senão
            F1 ← 0
            F2 ← 1
            para i de 1 até n - 1 faça
                  F ← F1 + F2
                  F1 ← F2
                  F2 ← F
            fimpara
            retorne F
      fimse
fimfuncao
```

Recursividade em C

- Utilizar a sintaxe de comandos e funções em C.

Exemplo em C


```c
#include <stdio.h>

int fatorial(int); /* definicao do prototipo */

void main ()
{
    int num;
    scanf("%d", &num);
    printf("Fatorial(%d): %d", num, fatorial(num));
}

int fatorial(int n)
{
    if (n ==0)
        return 1;
    else
        return n * fatorial(n-1);
}
```

Exercícios de Fixação

1) Escreva um algoritmo recursivo que lê um valor inteiro positivo n e faça a soma de todos os números inteiros positivos menores que o número digitado.

Exercícios de Fixação

2) Considere a seguinte série:

$$S = \frac{1}{1} + \frac{2}{3} + \frac{3}{5} + \frac{4}{7} + \cdots + \frac{20}{39}$$

Para achar a solução dessa série escreva uma função recursiva.