



# SCC120 - Capítulo 8

## Funções Recursivas em C

João Luís Garcia Rosa

Instituto de Ciências Matemáticas e de Computação  
Universidade de São Paulo - São Carlos  
<http://www.icmc.usp.br/~joaoluis>  
2010

1

### Recursão

- A recursão ocorre quando uma função chama a si própria. Quando isto acontece, várias ações ocorrem:
  - A função começa a execução do seu primeiro comando cada vez que é chamada;
  - Novas e distintas cópias dos parâmetros passados por valor e variáveis locais são criadas;
  - A posição que chama a função é colocada em estado de espera, enquanto que o nível gerado recursivamente esteja executando.
- O próximo programa explica estes efeitos.

2

# Contador recursivo

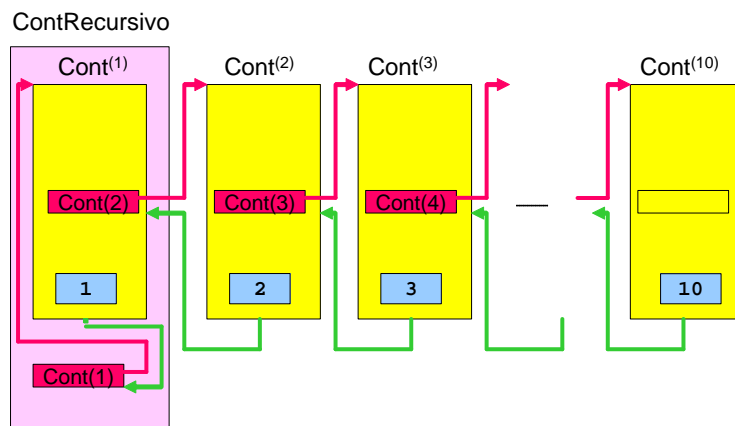
```
#include <stdio.h>

void cont (int);

main()
{
    cont(1);
}

void cont(int n)
{
    if (n < 10)
        cont(n+1);
    printf("%d\n", n);
}
```

3



4

# Fatorial

```
#include <stdio.h>

unsigned long fat(int);

main()
{
    short N;

    printf("Calculo do fatorial\n\n");
    printf("Entre com um numero inteiro: ");
    scanf("%d", &N);

    printf("\nO fatorial de %d %c %d\n\n", N, 130, fat(N));
}

unsigned long fat(int n)
{
    int fato = 1;
    if (n > 1)
        fato = n * fat(n-1);
    return fato;
}
```

5

```
#include <stdio.h>

void contreg(int n);

main()
{
    contreg(4);          // chama a função recursiva
}

void contreg(int n)
{
    printf("Contagem regressiva ... %d\n", n);
    if (n > 0)
        contreg(n-1);   // função chama a si mesmo
    printf("%d: Fogo!\n", n);
}
```

Saída:

```
Contagem regressiva ... 4
Contagem regressiva ... 3
Contagem regressiva ... 2
Contagem regressiva ... 1
Contagem regressiva ... 0
0: Fogo!
1: Fogo!
2: Fogo!
3: Fogo!
4: Fogo!
```

6

```

#include <stdio.h>
#define Comp 66
const int Divs = 6;

void subdivide(char ve[], int baixo, int alto, int nivel);

main()
{
    char regua[Comp];
    int i, j;
    int max = Comp - 2;
    int min = 0;
    for (i = 1; i < Comp - 2; i++)
        regua[i] = ' ';
    regua[Comp - 1] = '\0';
    regua[min] = regua[max] = '|';
    printf("%s\n", regua);
    for (i = 1; i <= Divs; i++)
    {
        subdivide(regua,min,max, i);
        printf("%s\n", regua);
        for (j = 1; j < Comp - 2; j++)
            regua[j] = ' '; // limpa a régua (branco)
    }
}

```

7

```

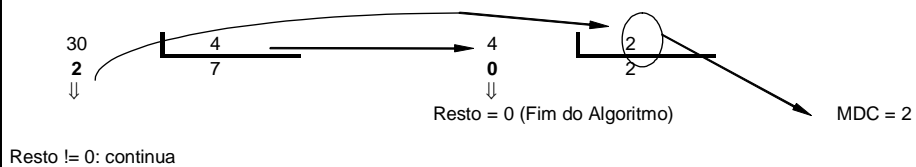
void subdivide(char ve[], int baixo, int alto, int nivel)
{
    int meio = (alto + baixo) / 2;
    if (nivel == 0)
        return;
    ve[meio] = '|';
    subdivide(ve, baixo, meio, nivel - 1);
    subdivide(ve, meio, alto, nivel - 1);
}

```

Saída:

8

**PROGRAMA PROPOSTO.** Escreva uma função recursiva que calcule o MDC (máximo divisor comum) de 2 números  $a$  e  $b$  recebidos como parâmetros. Para o cálculo do MDC, deve-se usar o Algoritmo de Euclides: Ex:  $a = 30$  e  $b = 4$ :



9

```
#include <stdio.h>

int MDC(int, int);

main()
{
    int a,b;
    printf("Calculo do MDC\n");
    printf("Entre com o valor de a: ");
    scanf("%d", &a);
    printf("Entre com o valor de b: ");
    scanf("%d", &b);
    printf("O MDC entre %d e %d %c %d\n", a, b, 130, MDC(a,b));
}

int MDC (int i, int j)
{
    if (i % j != 0)
        return MDC(j, i % j);
    else
        return j;
}
```

10