

## Computação Básica

Atividades a serem desenvolvidas nas sessões de Laboratório

### Sessão 13:

#### Objetivos:

- Exercitar a elaboração de programas que utilizam **funções recursivas**.

#### Atividades:

Em todos os programas abaixo, no corpo principal faça apenas chamadas de subalgoritmos.

1. Crie e execute o seguinte programa contendo a função X:

```
#include <stdio.h>

int X(int n, int m)
{
    if ((n==m) || (n==0))
        return 1;
    else
        return n + X(n-1,m);
}

int main()
{
    printf("Resultado da funcao X: %d ",X(5,3));
    getchar();
}
```

- a) Qual o valor de  $x(5,3)$  ?
- b) Quantas vezes a função X será chamada recursivamente?
- c) Insira no programa acima as linhas de comandos abaixo que estão em negrito. Compile, execute o programa e observe a saída.

```

#include <stdio.h>

int X(int n, int m)
{
    printf("Entrou na funcao com n= %d e m=%d",n,m);
    getchar();
    if ((n==m) || (n==0))
    {
        printf("Os valores de n e m se igualaram. Parou de chamar ela mesmo e retornou 1.");
        getchar();
        getchar();
        return 1;
    }
    else
    {
        printf("A funcao vai chamar ela mesmo, com n=%d e m=%d",n-1,m);
        getchar();
        return n + X(n-1,m);
    }
}

int main()
{
    printf("Resultado da funcao X: %d ",X(5,3));
    getchar();
}

```

2. A função abaixo calcula  $n!$ , sendo dado o valor de  $n$ , recursivamente:

```

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

```

- a) Elabore um programa que leia um número inteiro, maior ou igual a 0, e imprima  $n!$  utilizando a função acima. Execute o programa para diversos valores.
3. Escreva uma função recursiva que calcula e retorna a função potência  $x^y$ , onde o usuário deve informar valores para  $x$  e  $y$ .
4. Crie uma função recursiva para retornar o Máximo Divisor Comum (MDC) de dois números inteiros:
  - A definição do MDC é:
 

$MDC(x,y) = y$	se $x \geq y$ e $x \bmod y = 0$
$MDC(x,y) = MDC(y,x)$	se $x < y$
$MDC(x,y) = MDC(y, x \bmod y)$	caso contrário
  - Obs: mod é o resto da divisão, ou seja, % em C.

Dados para teste:  
 $MDC(12,18) = 6$   
 $MDC(45,15) = 15$   
 $MDC(46,6) = 2$   
 $MDC(560,240) = 80$

5. A soma dos primeiros N números inteiros pode ser calculado facilmente com um programa utilizando os laços (loops). Como poderíamos fazer a mesma soma, usando a recursão? Faça um função recursão que calcula a soma dos primeiros N números.

Dados para teste:	entrada:	saída:
	120	7260
	45	1035
	5	15
	7	28

6. Considere a definição para a série de Fibonacci:

$$F_0=0$$

$$F_1=1$$

$$F_n=F_{n-1} + F_{n-2}$$

Crie uma função recursiva para resolver a série de Fibonacci para um número qualquer.

Dados para teste:	entrada:	saída:
	4	3
	7	13
	10	55
	20	6765
	30	832040

OBS: A versão recursiva da série de Fibonacci tem uma natureza bastante explosiva. O número de chamadas para determinar  $F_n$  é igual ao número de chamadas para determinar  $F_{n-1}$  mais o número de chamadas para determinar  $F_{n-2}$  mais 1 para  $n>1$ . Para  $n=0$  ou  $n=1$  o número de chamadas é 1.

<b>valor de n</b>	<b>10</b>	<b>20</b>	<b>30</b>
<b>número de chamadas</b>	<b>177</b>	<b>21891</b>	<b>2692537</b>

Após implementar a sua função, apenas para finalidade de teste e curiosidade, coloque um contador dentro da mesma para mostrar quantas vezes ela foi chamada.