Lista de Exercícios de Programação Estruturada
Resursividade

1ª Questão: Execute o programa abaixo **manualmente**, mostrando a **pilha de execução**, conforme exemplos feitos em sala. Execute três vezes: a primeira vez com 1 disco, a segunda vez com 2 discos e a terceira vez com 3 discos.

```
#include <stdio.h>
void hanoi(int n, char a, char b, char c){
     if (n==1) {
         printf("Mova disco 1 de %c para %c \n", a, c);
     }
     else {
         hanoi(n-1,a,c,b);
         printf("Mova disco %d de %c para %c \n", n, a, c);
         hanoi(n-1,b,a,c);
      }
}
int main ( ) {
    int nDiscos;
    printf ("Informe a quantidade de discos (1 a 64): ");
    scanf ("%d", &nDiscos);
    hanoi (nDiscos, 'A', 'B', 'C');
    return 0;
 }
```

2ª Questão: O programa abaixo, apresentado em sala, exibe números em ordem decrescente de **num** até 1. Altere o programa abaixo, de forma que ele passe a exibir os números em ordem crescente de 1 até **num**. **Restrição**: o código precisa continuar recursivo.

```
#include <stdio.h>
void exibeDecrescente (int n) {
     if (n > 0) {
           printf ("%d ", n);
           exibeDecrescente (n-1);
     }
 }
 int main ( ) {
        int num;
        printf ("Informe um número inteiro: ");
        scanf ("%d", &num);
        exibeDecrescente (num);
        printf ("\n");
        return 0;
 }
```

3ª Questão: Implemente um programa para solicitar ao usuário um número N inteiro, maior que zero, e calcular e exibir a soma dos números inteiros de 1 a N. O programa deve FORÇAR que o usuário forneça um valor para N maior que zero. O programa deverá ser composto por dois métodos: o método main e uma **função recursiva** que irá calcular a soma. A função deverá receber N como parâmetro e retornar a soma calculada. O main deverá receber o N fornecido pelo usuário, chamar a função para calcular a soma e exibir a soma calculada pela função. Dica: soma (n) = n + soma (n-1).

4ª Questão: Implemente um programa para solicitar ao usuário dois números X (real) e Y (inteiro e maior ou igual a zero) e calcular X^Y. Para calcular X^Y, o programa deve utilizar uma **função recursiva**. Lembre-se que:

$$POWER(X,Y) = \begin{cases} 1, se Y = 0 \\ X, se Y = 1 \\ X * POWER(X, Y-1) \end{cases}$$

5ª Questão: Implemente um programa para solicitar ao usuário dois números inteiros e calcular seu máximo divisor comum. Para o cálculo do MDC, o programa deve utilizar uma **função recursiva**. Sabe-se que para MDC de dois números X e Y temos:

MDC(x,y) = MDC(x-y,y), se x > y
MDC(x,y) = MDC (y,x)
MDC(x,x) = x

Exemplo: MDC(12,4) = MDC(8,4) = MDC(4,4) = 4

6ª Questão: Implemente um programa para gerar e imprimir N termos da série de Fibonacci. O valor de N deve ser fornecido pelo usuário.

Série de Fibonacci => 1 1 2 3 5 8 13 21 34 55 89 ....

DICA: A série de Fibonacci pode ser definida recursivamente por:

$$fib(n) = \begin{cases} 1, n = 1, n = 2 \\ fib(n-1) + fib(n-2), n > 2 \end{cases}$$

O programa deve conter os seguintes sub-programas:

a) Uma **função recursiva** que gere o termo de ordem N da série de Fibonacci;
b) Um procedimento não recursivo que, utilizando a função definida em (a), gere a série de Fibonacci até o termo de ordem N.