

**Lista Prática 03**  
**Paradigma Imperativo (Linguagem Pascal)**

**Estrutura de uma *procedure***

(tipo de função que não devolve valor algum)

```
procedure nome (parametro(s):tipo1; parametro(s):tipo2);  
var  
    // declaração de variáveis  
begin  
    // comandos  
end;
```

Exemplo:

```
procedure EncontraMenor(x, y, z: integer; var m: integer);  
{encontra o menor valor entre três números inteiros}  
begin  
    if x < y then  
        m := x  
    else  
        m := y;  
    if z < m then  
        m := z;  
end; { fim do procedimento EncontraMenor }
```

**A variável m é passada por referência no procedimento EncontraMenor.**

**Estrutura de uma *function***

(tipo de função que devolve valor algum)

```
function nome (parametro(s):tipo1; parametro(s):tipo2):  
    tipo_de_retorno;  
var  
    // declaração de variáveis  
begin  
    // se vai devolver algum valor tem que fazer o nome receber o  
    retorno  
    // Exemplo: nome:=x+y;  
end;
```

Exemplo:

```
function EncontraMenor(x, y, z: integer):integer;  
{encontra o menor valor entre três números inteiros}  
var  
    m : integer;  
  
begin  
    if x < y then  
        m := x
```

```

else
    m := y;
if z < m then
    m := z;
EncontraMenor := m;
end; {fim da funcao EncontraMenor}

```

### Problema 1

Converta o seguinte programa em C para linguagem Pascal. Modularize-o utilizando procedimentos e funções.

```

int main()
{
    int b, e, res=1;
    scanf("%d",&b);
    scanf("%d",&e);
    for (int i=0;i<e;i++)
        res = res * b;
    printf("\n o resultado eh = %d \n\n",res);
    system("PAUSE");
    return 0;
}

```

### Problema 2

Elabore um programa que calcule uma divisão entre dois números inteiros e positivos utilizando uma sequência de subtrações. Deve-se considerar que o primeiro número (N1) será sempre maior que o segundo (N2), e que a divisão produz resultados exatos. Por exemplo: N1=10, N2=2, Resultado 5.

### Problema 3

Escreva uma função recursiva em Pascal para calcular o fatorial de um número.