1) Recursividade

Uma função que chama a si mesma, direta ou indiretamente, é dita ser recursiva. A técnica recursiva permite a escrita de algoritmos mais claros e precisos, especialmente problemas que possuem natureza recursiva.

O código abaixo mostra uma função recursiva em linguagem C para calcular a potência de um número inteiro:

```
#include <stdio.h>

int potencia(int base, int expoente){

  if(expoente == 0)
    return 1;
  else
    return (base * potencia(base, expoente-1));
}


int main(){
  int base = 2, expoente = 8, resultado;

  resultado = potencia(base,expoente);
  printf("%d elevado a %d = %d", base, expoente,resultado);

  return 0;
}
```

Em todas as funções recursivas existe:

- Um caso base, ou condição de parada, cujo resultado é imediatamente conhecido.
- Um passo recursivo em que se tenta resolver um sub-problema do problema inicial.

Exercícios em linguagem C:

1) O programa abaixo implementa duas funções para multiplicar dois números, sendo uma iterativa e outra recursiva. Analise as duas chamadas no programa principal e informe passo a passo o resultado obtido em cada uma.

```
#include <stdio.h>

long int mult(int x, int y){
   long int res=0;
   while( y != 0){
     res += x;
     y--;
   }
   return(res);
}
```

```
long int multRec(int x, int y)
{
   if (y == 0)
      return 0;
   else
      return(x + multRec(x, y-1));
}

int main(){
    printf("\nResultado Iterativo: %d",mult(5,600));
    printf("\nResultado Recursivo: %d\n\n",multRec(5,600));
    system("pause");
    return 0;
}
```

2) Seja a somatória abaixo. Faça uma função recursiva para realizar o cálculo.

$$\sum_{i=1}^{n} = i^2$$

3) A função de *Ackermann* é definida para valores inteiros e não negativos *m* e *n* da seguinte forma:

$$A(m,n) = \begin{cases} n+1 & \text{se } m = 0 \\ A(m-1,1) & \text{se } m > 0 \text{ e } n = 0 \\ A(m-1, A(m,n-1)) & \text{se } m > 0 \text{ e } n > 0. \end{cases}$$

Faça uma função recursiva para implementá-la. Qual o valor de A(3,2)?

4) Escreva uma função recursiva que calcule o número de grupos distintos com k pessoas que podem ser formados a partir de um conjunto de n pessoas. A definição abaixo da função ***Comb(n,k)*** define as regras:

$$Comb(n,k) = \begin{cases} n & \text{se } k = 1 \\ 1 & \text{se } k = n \\ Comb(n-1,k-1) + Comb(n-1,k) & \text{se } 1 < k < n \end{cases}$$