



# Algoritmos e Programação II

**Professora:** Graziela Araújo

## *Exercícios de recursão com C*

1. Mostre, através de teste de mesa, o resultado das seguintes funções:

```
int f1(int n)
{
    if (n == 0)
        return (1);
    else
        return(n * f1(n-1));
}
```

Considere as entradas:

- I. f1(0);
- II. f1(1);
- III. f1(5);

```

int f2(int n)
{
    if (n == 0)
        return (1);
    if (n == 1)
        return (1);
    else
        return(f2(n-1)+ 2 * f2(n-2));
}

```

**Considere as entradas:**

I. f2(0);

II. f2(1);

III. f2(5);

```

int f3(int n)
{
    if (n == 0)
        printf("Zero ");
    else
    {
        printf("%d ",n);
        printf("%d ",n);
        f3(n-1);
    }
}

```

**Considere as entradas:**

I. f3(0);

II. f3(1);

III. f3(5);

2. Determine o que a seguinte função recursiva em C calcula. Escreva uma função iterativa para atingir o mesmo objetivo.

```

func (int n)
{
    if (n == 0)
        return(0);

    return(n + func(n-1));
}

```

3. Os números de Pell são definidos pela seguinte recursão

$$p(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ 2p(n-1) + p(n-2) & \text{se } n > 2 \end{cases}$$

Alguns números desta sequência são: 0, 1, 2, 5, 12, 29, 70, 169, 408, 985...  
E os números de Catalan são definidos pela seguinte recursão

$$C(n) = \begin{cases} 1 & \text{se } n = 0 \\ \frac{2(2n-1)}{n+1} C(n-1) & \text{se } n > 0 \end{cases}$$

Alguns números desta sequência são: 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786...

Faça duas funções recursivas que recebam um número N e retorne o N-ésimo número de: Pell e Catalan.

4. Um palíndromo é uma string que é lida da mesma maneira da esquerda para a direita e da direita para a esquerda. Alguns exemplos de palíndromo são radar e a bola da loba (se os espaços forem ignorados) Escreva uma função recursiva que retorna 1 se a string armazenada no array for um palíndromo e 0, caso contrário. O método deve ignorar espaços e pontuação na string.
5. Faça uma função recursiva que permita calcular a média de um vetor de tamanho N.
6. Defina a função soma\_nat que recebe como argumento um número natural n e devolve a soma de todos os números naturais até n.
7. Defina a função prim\_alg que recebe como argumento um número natural e devolve o primeiro algarismo (o mais significativo) na representação decimal de n.
8. Defina a função num\_perf que recebe como argumento um número inteiro positivo e devolve True se esse número for um número perfeito e False em caso contrário. Recorde que um número perfeito é um número natural que é igual à soma de todos os seus divisores próprios, isto é, a soma de todos os divisores excluindo o próprio número. Pode, se assim o entender, definir funções auxiliares.
9. Defina a função conta que recebe como argumentos uma lista de números inteiros w e um número inteiro k e devolve o número de vezes que k ocorre em w.
10. Defina a função lposicoes que recebe como argumentos uma lista de números inteiros w e um número inteiro k e devolve a lista das posições em que k ocorre em w.