AULA 8 - ANÁLISE DA COMPLEXIDADE DE ALGORITMOS RECURSIVOS (NÚMERO DE PERRIN)

- Implemente uma **função recursiva** para calcular os números de Perrin

  3, 0, 2, 3, 2, 5, 5, 7, 10, 12, 17, 22, 29, 39, ... (https://oeis.org/A001608)

  definidos pela seguinte relação de recorrência:

$$P(n) = \begin{cases} 3, se\ n = 0 \\ 0, se\ n = 1 \\ 2, se\ n = 2 \\ P(n-2) + P(n-3), \text{se } n > 2 \end{cases}$$

  - Construa um programa para executar a função para sucessivos valores de n e que permita determinar experimentalmente a ordem de complexidade das operações de soma do seu algoritmo. Efetue a **análise empírica** da complexidade construindo uma tabela com o número de adições efetuadas para diferentes valores de n. Qual é a **ordem de complexidade** da função recursiva?

Uma forma de resolver problemas recursivos de maneira a evitar o cálculo repetido de valores, consiste em calcular os valores de baixo para cima, ou seja, de Perrin (0) para Perrin (n) e utilizar um *array* para manter os valores entretanto calculados. Este método designa-se por programação dinâmica e reduz o tempo de cálculo à custa da utilização de mais memória para armazenar valores intermédios.

- Usando a técnica de **programação dinâmica**, implemente uma função repetitiva alternativa e efetue a análise empírica da sua complexidade. Qual é a **ordem de complexidade** da função repetitiva?

Escreva o código das funções no verso da folha

N	Recursivo (N)	Nº de Adições	Prog. Din. (N)	Nº de Adições
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
O(N)				

- **Determine formalmente a ordem de complexidade dos dois algoritmos, obtendo expressões matemáticas exatas e simplificadas.**

NOME: **Nº MEC:**

NOME: **Nº MEC:**