

Sección 1: La secuencia Fibonacci puede definirse de la siguiente manera.

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_n &= F_{n-1} + F_{n-2} \end{aligned}$$

Por lo que los primeros números de la secuencia son los siguientes:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ...

1. Diseña un algoritmo recursivo que calcule el n-ésimo elemento de la secuencia.

```
public static int fibonacciRecursive(int n) {  
    }  
}
```

2. Diseña un algoritmo iterativo que calcule el n-ésimo elemento de la secuencia.

```
public static int fibonacciIterative(int n) {  
    }  
}
```

3. ¿Cuál algoritmo es más eficiente? ¿Por qué? Justifica tu respuesta.

Sección 2: Diseña un método recursivo que verifique si una palabra es un palíndromo. Los palíndromos se definen como una palabra que se lee igual al derecho y al revés. Por ejemplo:

```
isPalindrome("tenet") → true  
isPalindrome("hola") → false  
isPalindrome("") → true
```

```
public static boolean isPalindrome(String word) {  
    }  
}
```

Sección 3. Realiza una búsqueda binaria sobre los siguientes arreglos. Muestra cada uno de los pasos. Indica los valores de begin, mid y end. Realiza el procedimiento a mano, o utiliza una hoja de cálculo (Excel) para representar cada paso.

1	2	3	7	300	2480	7855
---	---	---	---	-----	------	------

target = 2480

0	35	70	777	1000	1001	1020	3985	4031	5203	9811	99812	352330
---	----	----	-----	------	------	------	------	------	------	------	-------	--------

target = 777

-26	-11	0	9	51	61	76	85	390
-----	-----	---	---	----	----	----	----	-----

target = 0

-26	-11	0	9	51	61	76	85	390
-----	-----	---	---	----	----	----	----	-----

target = 100