

*En cada ejercicio se evaluará la eficiencia del código, el uso de identificadores significativos, la indentación, escritura correcta de llaves {} y el uso adecuado de la palabra reservada const. Se dispone de dos horas para entregar la prueba y debe realizarse en forma estrictamente individual.*

Un instituto de enseñanza de la matemática quiere implementar un juego que haga más divertido el aprendizaje de las sucesiones numéricas. Una sucesión numérica es un conjunto ordenado de números enteros que guardan un patrón entre ellos. A cada número de la sucesión se le llama elemento, término o miembro. Ejemplos de sucesiones son:

**Tabla 1.** Ejemplos de algunas sucesiones

Ej	Primeros elementos	Descripción	Definición recursiva	Definición iterativa
1	0, 2, 4, 6, 8, 10	números pares	$a_n = a_{n-1} + 2$ $a_0 = 0$	$a_n = 2n$
2	1, 2, 4, 8, 16, 32	potencias de dos	$a_n = 2a_{n-1}$ $a_0 = 1$	$a_n = 2^n$
3	1, 4, 9, 16, 25, 36	cuadrados		$a_n = n^2$
4	0, 1, 3, 6, 10, 15	progresión numérica de Gauss	$a_n = a_{n-1} + n$ $a_0 = 0$	$a_n = \frac{n(n+1)}{2}$
5	1, 1, 2, 6, 24, 120	factoriales	$a_n = na_{n-1}$ $a_0 = 1$	$a_n = 1 \cdot 2 \cdot 3 \dots \cdot n$
6	0, 1, 1, 2, 3, 5	sucesión de Fibonacci	$a_n = a_{n-1} + a_{n-2}$ $a_0 = 1, a_1 = 1$	
7	2, 3, 5, 7, 11, 13	números primos		
8	1, 8, 3, 6, 5, 4, 7, 2	impares crecientes intercalados con pares decrecientes		$a_n = n + 1$ si n es par $a_n = N - n + 1$ si n es impar

Para probar el concepto, en cada intento el juego debe escoger una secuencia aleatoria, generar algunos elementos, ocultar uno al azar, e imprimir los elementos indicando el oculto con guiones bajos. El jugador debe adivinar el número oculto. Ejemplo de ejecución:

```

./sequence 7
Sequence 1: 12, 14, 16, __, 20, 22, 24
0 points. Answer: 18
Correct!

Sequence 2: __, 27, 30, 33, 36, 39, 42
1 points. Answer: 24
Correct!

Sequence 3: 49, 64, __, 100, 121, 144, 169
3 points. Answer: 81
Correct!

Sequence 4: 38, 43, 39, __, 40, 41, 39
6 points. Answer: 42
Wrong! Expected 41

Sequence 5: 64, 128, 256, 512, 1024, ____, 4096
4 points. Answer: 2048
Correct!

Sequence 6: 4, 6, 1, 3, 5, __, 2
7 points. Answer: EOF
$

```

Cada secuencia tiene un nivel de dificultad. Es un número entero que distingue a una secuencia de otra por su nivel de dificultad. Entre mayor este número, mayor la dificultad. Cuando el jugador acierta una secuencia, el juego incrementa el nivel de dificultad alcanzado por el jugador. Si el jugador falla, el juego decrementa este nivel de dificultad. El juego sólo lanza al jugador secuencias del mismo nivel de dificultad o inferior al nivel que ha alcanzado.

El jugador tiene un marcador. Cada vez que el jugador acierta una secuencia, se le acreditan tantos puntos como la dificultad de la secuencia. Si el jugador falla la secuencia, se le debitan puntos equivalentes a la mitad de la dificultad de la secuencia que falla.

El jugador puede terminar el juego en cualquier momento al introducir el carácter fin de archivo como respuesta. El jugador compete contra sí mismo, tratando de mejorar su marcador. En futuras versiones los jugadores podrían competir entre ellos.

Al invocar el juego desde la línea de comandos se puede proveer un número mayor o igual a 6 como parámetro. Indica la cantidad de elementos de cada secuencia que el juego debe imprimir. Si este número no se especifica, se asume el mínimo 6.

El instituto requiere que el juego sea extensible. Es decir, que sea fácil agregar nuevas secuencias en el futuro de las más de cien mil disponibles en [oeis.org](http://oeis.org). Evaluación:

1. [15%] Implementa una clase controladora que representa al juego como un todo. Se encarga del análisis de parámetros. Provee una función `main()`.
2. [15%] Implementa una estructura de datos para almacenar secuencias, y escogerlas aleatoriamente. Llena esa estructura apropiadamente.
3. [15%] Implementa la lógica del juego. Actualiza el marcador del jugador de acuerdo a sus aciertos o fallos. Escoge secuencias al azar acordes al nivel de dificultad alcanzado por el jugador.
4. [20%] Implementa una clase abstracta `Sequence` que impone la interfaz que toda secuencia debe cumplir, de tal forma que agregar una nueva no implica cambios para el resto de juego.
5. [10%] Implementa una secuencia concreta escogida entre las últimas cinco de la Tabla 1.
6. [15%] Implementa una secuencia concreta de dificultad 5 o superior que **no** esté en la Tabla 1.
7. [10%] Evita fugas de memoria a toda costa.