

Ejercicios Bucles '*Avanzados*'

[Descargar estos ejercicios](#)

Índice

1. [Ejercicio 1](#)
2. [Ejercicio 2](#)
3. [Ejercicio 3](#)
4. ☒ [Ejercicio 4](#)
5. [Ejercicio 5](#)
6. ☒ [Ejercicio 6](#)
7. ☒ [Ejercicio 7](#)
8. ☒ [Ejercicio 8](#)
9. [Ejercicio 9](#)
10. [Ejercicio 10](#)

Ejercicio 1

Mostrar un tablero de ajedrez y sobre él, marcar con `*` las celdas a las que se puede mover un alfil desde una posición **dada por el usuario**.

Ten en cuenta que en ajedrez las filas son numéricas del **1 al 8** y las columnas letras de la **A a la H**.

💡 **Tips:** Aunque no es necesario, si quieres imprimir colores en el terminal puedes usar las propiedades de consola:

```
Console.BackgroundColor = ConsoleColor.Color;  
Console.ForegroundColor = ConsoleColor.Color;
```

Además, puedes imprimir en una posición específica en pantalla con.

```
Console.SetCursorPosition(Columna, Fila);
```

Para más información sobre la clase [Console](#)

Un ejemplo de **salida por pantalla** puede ser el siguiente:

```
Introduce fila: 4  
Introduce columna: D  
ABCDEFGH  
1*      *  
2 *    *  
3  *  *  
4   A  
5  *  *  
6 *    *  
7*      *  
8       *
```

Ejercicio 2

Muestra un tablero de ajedrez y sobre él marcar con * las celdas a las que se puede mover un caballo desde una posición **dada por el usuario**.

Un ejemplo de **salida por pantalla** puede ser el siguiente:

```
Introduce fila: 4
Introduce columna: D
ABCDEFGH
1
2  * *
3 *   *
4   C
5 *   *
6 * *
7
8
```

Ejercicio 3

Construir un **triángulo de N filas** con números.

Un ejemplo de **salida por pantalla** puede ser el siguiente:

```
1
23
345
4567
56789
678901
```

✓ Ejercicio 4

Igual que el anterior pero para este otro triángulo.

Un ejemplo de **salida por pantalla** puede ser el siguiente:

```
3
58
703
9258
```

Ejercicio 5

Una progresión aritmética tiene el siguiente término general $a_n = 3 * n - 2$. Se desea un programa que pida un número k y calcule los **k primeros términos**.

Un ejemplo de **salida por pantalla** puede ser el siguiente:

```
k: 5
Progresión a1=1, a2=4, a3=7, a4=10, a5=13.
```

✓ Ejercicio 6

Programa la generación de una tabla de verdad con unos y ceros **con un bucle for anidado**.

- Las operaciones que mostrará la tabla serán por tanto **operaciones de bit**.
- Para ello el usuario deberá introducir por teclado un carácter con el cual nos indicará la tabla a generar (&, |, ^).
- La decisión de la tabla a mostrar, debe ser mediante una **instrucción switch** con el carácter.
- En el caso de tratarse de una operación no válida avisaremos de tal hecho y finalizaremos la ejecución.

Un ejemplo de **salida por pantalla** puede ser el siguiente:

```
Introduce una operación de bit (&, |, ^): &
0 & 0 = 0
0 & 1 = 0
1 & 0 = 0
1 & 1 = 1
```

✓ Ejercicio 7

Escribe un programa que genere la secuencia de números:

1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5, ..., 1, 2, 3, ... N

✓ Ejercicio 8

Un número se dice que es **capicúa** si leído de derecha a izquierda da el mismo resultado que leído de izquierda a derecha.

Por ejemplo, los números **22**, **343**, **5665** o **12321** son capicúas.

Elabora un programa que lea desde teclado un **número entero mayor de 9** y devuelva si el número es capicúa o no.

Ejercicio 9

Crea un programa que muestre en pantalla la siguiente pirámide:

```
  1
 232
34543
4567654
567898765
67890109876
```

Se introducirá por teclado un número que indique la profundidad de la pirámide.

Nota: Para rellenar de blancos cada línea nos fijaremos en la profundidad de la pirámide. Si la profundidad de la pirámide es **n** requeriremos **n-1** blancos para la **primera línea**, **n-2** para la **segunda**, **n-3** para la **tercera** y **así sucesivamente**.

Ejercicio 10

Haz un programa que usando algún tipo de bucle. Determine la ubicación de un número mayor que cero (leído del teclado) en una lista de números mayores que cero leída del teclado (lista creciente estrictamente y que finalizará con un 0).

Si suponemos la entrada de datos...

```
Entrada → (Número buscado) en (Lista)
```

Podemos obtener los siguientes ejemplos de solución...

```
2 en 3 5 6 7 8 0
Fuera de lista a la Izquierda

8 en 1 3 5 7 0
Fuera de lista a la Derecha

4 en 1 3 4 6 8 0
En la lista

5 en 1 3 4 7 0
Fuera de lista a la Intercalado
```

