Índice

Ejercicios Unidad 6

Descargar estos ejercicios

Antes de empezar

Para realizar estos ejercicios, deberás descargar los recursos del enlace de Descargar recursos. Cada ejercicio se codificará en el cuerpo del método que corresponda con el nombre del ejercicio. Es decir, sustituirás las líneas `//TODO:` por el código de solución del ejercicio. Para probar el funcionamiento correcto de los ejercicios, deberás pasar los Test adjuntos a este proyecto (como se explica en la unidad 4).

Ejercicio 1. Números amigos

Programa que determina si dos números enteros positivos son amigos. Dos números son amigos si la suma de los divisores del primer número excepto él mismo, es igual al segundo numero, y viceversa. Puedes saber un poco más de la historia de esta relación entre números leyendo la entrada en la Wikipedia.

```
Ejercicio 1: Contador de números
Introduzca valor 1: -3
Introduzca valor 2: 5
ERROR: Sólo se permiten números positivos

Introduzca valor 1: 3
Introduzca valor 2: 5

Los valores no son amigos
```

```
Introduzca valor 1: 220
Introduzca valor 2: 284

Los valores  son amigos
```

Requisitos:

- Usa bucles `for` o `while` para resolverlo.
- Controla que los números tienen que ser positivos.

Ejercicio 2. Calculadora

Realiza un programa que sea capaz de **sumar**, **restar**, **multiplicar** y dividir.

```
--- CALCULADORA ---
1. Sumar
2. Restar
3. Multiplicar
4. Dividir
ESC. Salir
Pulsa una opción: 3
Introduce el primer operando: 34
Introduce el segundo operando: 2
Resultado: 68
```

Requisitos:

- El programa presentará un menú con las cuatro operaciones que puede realizar.
- Saldrás del programa con la tecla **ESC**. Para controlar que se ha pulsado la tecla ESC podemos usar la instrucción Console.ReadKey() en un esquema similar a la siguiente propuesta dentro del bucle...

```
// ... Mostrar menú
Console.Write("Pulsa una opción: ");
var tecla = Console.ReadKey(); // Esta instrucción se para hasta que se pulse una tecla. (Sin intr

// Introducción de operandos ...
double o1 = ....;
double o2 = ....;

bool esEscape = tecla.Key == ConsoleKey.Escape; // Comprueba si la tecla pulsada es escape.

if (!esEscape)
{
    char caracterAsociadoALaTecla = tecla.KeyChar;  // Obtener el caracter asociado a la tecla.
    double resultado = caracterAsociadoALaTecla switch
    {
        '1' => o1 + o2,
        //...
        _ => double.NaN
    };

    // Mostrar resultado.
}
```

- Si el número de entrada de el menú no es válido, se mostrará la salida.

```
....
Introduce el segundo operando: 2
Operación no válida
....
```

Ejercicio 3. Multiplos de 7

Mostrar los **múltiplos de 7** que hay entre **7** y **112**.

```
Ejercicio 3. Múltiplos de 7

7 14 21 28 35 42 49 56 63 70 77 84 91 98 105
```

Requisitos:

- Usa un bucle `for`.

Ejercicio 4. Secuencia de números

Pide un número, por ejemplo el 4, y saca en pantalla 1223334444.

```
Ejercicio 4. Secuencia de números
Introduzca un numero entero: 8
1223334444555556666667777777888888888
```

Requisitos:

- Debes usar dos bucles `for` anidados

Ejercicio 5. Triángulo de asteriscos

Escribe un programa que dibuje un cuadrado de asteriscos de N×N, donde N es introducido por el usuario.

```
Ejercicio 5: Triángulo de asteriscos
Introduce el tamaño del triángulo: 4
*
* *
* * *
* * * *
```

Requisitos:

- Usa bucles `for` anidados.
- El bucle exterior controla las filas.
- El bucle interior controla las columnas.

Ejercicio 6. Pirámide de números

Crea un programa que dibuje una pirámide de números de N filas, donde N es introducido por el usuario.

```
Ejercicio 6: Pirámide de números
Introduce el número de filas: 5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Requisitos:

- Usa bucles `for` anidados.
- En cada fila i, se imprimen los números del 1 al i.

Ejercicio 7. Suma de dígitos

Lee un número y escribe la suma de sus dígitos.

```
Ejercicio 7. Suma de dígitos
Introduzca un numero: 1234
Número de dígitos: 4
Suma de dígitos: 10
```

Requisitos:

- No puedes usar **arrays** ni cadenas.
- Usa bucle `while` anidados.
- Divide entre 10 para descomponer el número y usa el `%` para coger los restos.

Ejercicio 8. Pirámide de números dos

Crea un programa que dibuje una pirámide de números de N filas, donde N es introducido por el usuario.

```
Ejercicio 8: Pirámide de números dos
Introduce el número de filas: 5
3
5 8
7 0 3
9 2 5 8
```

Requisitos:

- Usa bucles `for` anidados.
- El primer número de cada fila sigue una cuenta creciente de números impares empezando en 3
- Los números en la columna se incrementan de 3 en 3 a partir del primer de la fila.
- Cuando la cuenta supera 10 se visualiza el `%` de dividir por 10.

Ejercicio 9. Pirámide de números completa

Crea un programa que muestre en pantalla la siguiente pirámide:

```
Ejercicio 9. Pirámide de números completa
Introduce el número de filas: 6
     1
    232
   34543
  4567654
 567898765
67890109876
```

Requisitos:

- Usa bucles `for` anidados.
- Se introducirá por teclado un número que indique la profundidad de la pirámide.
- Cuando la cuenta supera 10 se visualiza el `%` de dividir por 10.
- 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. Buscar la ubicación de un número

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).
Para cuatro entradas distintas de datos, podemos tener las siguientes salidas...

```
Ejercicio 10. Buscar la ubicación de un número

Ejercicio 10: Buscar la ubicación de un número
Introduce el número a ubicar: 5
Introduce los números de la lista en orden ascendente(terminando con 0):
1
3
4
0
Fuera de lista a la Derecha
```

Requisitos:

- Pide el número a ubicar
- Usa un bucle `while` para ir pidiendo los números de la lista (el usuario los tendrá que introducir de forma creciente).
- Para cada número introducido se comprueba con el de ubicar y se prepara el texto de salida que podrá ir cambiando con el resto de entradas.
- Se acabará con un 0 o cuando se introduzca el número a situar.
- Otras ejemplos de entrada:

```
2 en 3 5 6 7 8 0
Fuera de lista a la Izquierda
....
4 en 1 3 4 6 8 0
En la lista
....
5 en 1 3 4 7 0
Fuera de lista a la Intercalado
```