



Ejercicios

Introducción a .NET y C#

Ejercicios básicos en C#

Antes de empezar

- Deberá asignarle a la propiedad 'Title' de la clase Console, el número de ejercicio.

Por ejemplo:

Console.Title = "Ejercicio Nro ##" donde ## será el número del ejercicio.

- Del mismo modo se deberán nombrar los proyectos de consola.

Por ejemplo:

Ejercicio_##.

- Para visualizar los valores decimales de los ejercicios, deberá dar el siguiente formato de salida al método Write/WriteLine: "#,###.00".

I01. Máximo, mínimo y promedio

Ingresar 5 números por consola, guardándolos en una variable escalar. Luego calcular y mostrar: el valor máximo, el valor mínimo y el promedio.

*Nota: Un **escalar** es una constante o variable que contiene un dato atómico y unidimensional. En contraposición al concepto de escalar, están los conceptos de array, lista y objeto, que pueden tener almacenado en su estructura más de un valor.*

I02. Error al cubo

Ingresar un número y mostrar el cuadrado y el cubo del mismo. Se debe validar que el número sea mayor que cero, caso contrario, mostrar el mensaje: "ERROR. ¡Reingresar número!".

*Nota: Utilizar el método **Pow** de la clase **Math** para calcular las potencias.*

I03. Los primos

Mostrar por pantalla todos los números primos que haya hasta el número que ingrese el usuario por consola.

Validar que el dato ingresado por el usuario sea un número.

Volver a pedir el dato hasta que sea válido o el usuario ingrese "salir".

Si ingresa "salir", cierra la consola.

Al finalizar, preguntar al usuario si desea volver a operar. Si la respuesta es afirmativa, iterar. De lo contrario, cerrar la consola.

Nota: Utilizar sentencias de iteración, selectivas y el operador módulo (%).

I04. Un número perfecto

Un número perfecto es un entero positivo, que es igual a la suma de todos los enteros positivos (excluido el mismo) que son divisores del número.

El primer número perfecto es 6, ya que los divisores de 6 son 1, 2 y 3; y $1 + 2 + 3 = 6$.

Escribir una aplicación que encuentre los 4 primeros números perfectos.

Nota: Utilizar sentencias de iteración y selectivas.

I05. Tírame un centro

Un centro numérico es un número que separa una lista de números enteros (comenzando en 1) en dos grupos de números, cuyas sumas son iguales.

El primer centro numérico es el 6, el cual separa la lista (1 a 8) en los grupos: (1; 2; 3; 4; 5) y (7; 8) cuyas sumas son ambas iguales a 15. El segundo centro numérico es el 35, el cual separa la lista (1 a 49) en los grupos: (1 a 34) y (36 a 49) cuyas sumas son ambas iguales a 595.

Se pide elaborar una aplicación que calcule los centros numéricos entre 1 y el número que el usuario ingrese por consola.

Nota: Utilizar sentencias de iteración y selectivas.

I06. Años bisiestos

Escribir un programa que determine si un año es bisiesto.

Un año es bisiesto si es múltiplo de 4. Los años múltiplos de 100 no son bisiestos, salvo si ellos también son múltiplos de 400. Por ejemplo: el año 2000 es bisiesto, pero 1900 no.

Pedirle al usuario un año de inicio y otro de fin y mostrar todos los años bisiestos en ese rango.

Nota: Utilizar sentencias de iteración, selectivas y el operador módulo (%).

I07. Recibo de sueldo

Se debe pedir el ingreso por teclado del valor hora, el nombre, la antigüedad (en años) y la cantidad de horas trabajadas en el mes de **N** cantidad de empleados de una fábrica.

Se pide calcular el importe a cobrar teniendo en cuenta que el total (que resulta de multiplicar el valor hora por la cantidad de horas trabajadas), hay que sumarle la cantidad de años trabajados multiplicados por \$150, y al total de todas esas operaciones restarle el 13% en concepto de descuentos.

Mostrar el recibo correspondiente con el *nombre*, la *antigüedad*, el *valor hora*, el *total a cobrar en bruto* y el *total a cobrar neto* de todos los empleados ingresados.

Nota: Utilizar sentencias de iteración y selectivas. No es necesario ni se deben utilizar vectores/arrays.

I08. Dibujando un triángulo rectángulo

Escribir un programa que imprima por consola un triángulo como el siguiente:

```
*
***
*****
*****
*****
```

El usuario indicará cuál será la altura del triángulo ingresando un número entero positivo. Para el ejemplo anterior, la altura ingresada fue de 5.

Nota: Utilizar sentencias de iteración y selectivas.

I09. Dibujando un triángulo equilátero

Escribir un programa que imprima por consola un triángulo como el siguiente:

```
  *
 ***
*****
*****
*****
```

El usuario indicará cuál será la altura del triángulo ingresando un número entero positivo. Para el ejemplo anterior, la altura ingresada fue de 5.

Nota: Utilizar sentencias de iteración y selectivas.