Ejercicios Métodos 'Avanzados'

Descargar estos ejercicios

Índice

Ejercicio 1

> Escribe un método al se le pase un número y diga **si es primo o no**. Suponiendo ya definido el método anterior, escribe un programa que lea dos números enteros por teclado y los sume sólo si son primos, indicando si el resultado también es primo.
> En caso contrario, debe decir cuál (o cuáles) de ellos no son primos.

✅ Ejercicio 2

Realiza un programa para calcular el número mágico de una persona. El número mágico de una persona debe considerarse como la suma de las cifras de su día de nacimiento, repitiendo el proceso hasta que la suma de las cifras devuelva un número menor de 10.
De esta forma, alguien nacido el 7 de marzo de 1965 tendría como número mágico el 4.

👁 **Nota:** Deberás usar los métodos necesarios para que tu programa quede bien modularizado y con alta cohesión, evitando el acoplamiento.

Un ejemplo de ejecución podría ser:

```
Día nacimiento: 7
Mes nacimiento: 3
Año nacimiento: 1965
7 + 3 + 1 + 9 + 6 + 5 = 31
3 + 1 = 4
Tu número mágico es: 4
```

Ejercicio 3

Escribir en forma de **tabla** los cuadrados y los cubos de los 10 primeros números enteros positivos. Definir para ello un método **Cuadrado** y otra **Cubo**, que te eleven un número al cuadrado y otro al cubo.

Un ejemplo de ejecución podría ser:

```
Numero Cuadrado Cubo
1       1       1
2       4       8
3       9       27
...
10      100     1000
```

Ejercicio 4

Escribe el resultado de ejecutar este programa y comenta el motivo de la salida.

👁 **Nota:** Puedes ayudarte con la traza.

```
class Program
{
    static public int a = 0, b = 3;

    static int Funcion1(int a)
    {
        b = b + Funcion2(a) + 1;
        return b;
    }
    static int Funcion2(int a)
    {
        return (b + a + 1);
    }
    static void Main()
    {
        int cont;
        for (cont = 1; cont <= 5; cont++)
        {
            b = Funcion1(cont + a + b) + 1;
            Console.WriteLine("{0} ", b);
        }
    }
}
```

☑ Ejercicio 5

Escribe un programa que lea un número en base 10 (decimal) y que posteriormente muestre un menú que nos permita convertirlo a base 2 (binario), base 8 (octal) o base 16 (hexadecimal).

Crea un método llamado **PasaABinario**, otro llamado **PasaAOctal** y otro llamado **PasaAHexadecimal** para realizar dichas operaciones.
Todos los métodos **devolverán un string** que contendrá los dígitos resultantes de la conversión.

👁 **Nota 1:** No se puede usar el método **Convert** existente ya en C#.
👁 **Nota 2:** Evita el código repetido.

🔍 **Algunas pistas para resolver el ejercicio:**

1. Si no sabes como es el proceso de conversión matemático, puedes consultar el proceso de decimal a binario en el siguiente enlace: Decimal a binario
   Para el resto de sistemas es análogo solo que dividiendo entre 8 y 16 respectivamente.
2. En todos los casos obtendremos restos **enteros** entre **0 a 15** que deberemos pasar a cadena o a caracter.
   Una de las opciones es pasar a cadena. En este caso cuando el valor esté entre **0 a 9** generaremos una cadena con **"0", "1" ... o , "9"** por ejemplo con **$"{resto}"**. Sin embargo, cuando el valor este entre **10 y 15**, generaremos una cadena con los caracteres **"A", "B", ... o ,"F"** y eso se puede hacer, por ejemplo, con un expresión switch.
   De tal manera que podríamos tener una expresión del tipo ...

```
string textoResto = resto < 10
    ? $"{resto}"
    : resto switch { 10 => "A", 11 => "B", 12 => "C", 13 => "D", 4 => "E", 15 => "F", _ => "?" };
```

   Mas **eficiente** y tradicional de programadores de C y C++ es usar la tabla UTF-8 o su subconjunto ASCII para en lugar de pasar a cadena pasar a **carácter**. El planteamiento sería el siguiente:
   - El carácter '0' internamente se guarda con el valor 48 (Decimal) y el resto de caracteres que representan dígitos decimales **van seguidos en esa tabla**. Por tanto, si cuando el resto es menor que 10 hago...

```
const int INCIO_CARCACTER_0 = 48;
char simbolo = (char)(INCIO_CARCACTER_0 + resto);
```

   - El carácter mayúscula 'A' se guarda internamente con el valor 65 y la 'B', 'C',..., 'F' van seguidos en esa tabla luego...

   Puedes utilizar este último planteamiento si lo consideras más adecuado.

3. En el proceso de crear la cadena con la conversión, igual se te plantea la necesidad de invertir una cadena. Veamosló con un ejemplo...

```
Supongamos que queremos pasar el 6 en decimal a binario (110)

En algún momento inicializarás la cadena de conversión.
cadenaConLaConversion = "";

En laa iteración 1
6 % 2 = 0 entonces cadenaConLaConversion += simboloDelResto;
Ahora cadenaConLaConversion = "0" y obtendremos el cociente para siguiente iteración con 6 / 2 = 3

En la iteración 2
3 % 2 = 1 entonces cadenaConLaConversion += simboloDelResto;
Ahora cadenaConLaConversion = "01" y obtendremos el cociente para siguiente iteración con 2 / 2 = 1

Como cociente < base entonces cadenaConLaConversion += simboloDelCociente;
Ahora cadenaConLaConversion final valdrá "011"
```

Para tener la conversión correcta "011" os puede surgir la necesidad de invertir la cadena para obtener "110" que es el 6 en binario. De momento solo sabemos concatenarlas con el operador + o += por tanto una forma de resolver esto sería la siguiente ...

```
En laa iteración 1
6 % 2 = 0 entonces cadenaConLaConversion = simboloDelResto + cadenaConLaConversion;
Ahora cadenaConLaConversion = "0" y obtendremos el cociente para siguiente iteración con 6 / 2 = 3

En la iteración 2
3 % 2 = 1 entonces cadenaConLaConversion = simboloDelResto + cadenaConLaConversion;
Ahora cadenaConLaConversion = "10" y obtendremos el cociente para siguiente iteración con 2 / 2 = 1

Como cociente < base entonces cadenaConLaConversion = simboloDelCociente + cadenaConLaConversion;
Ahora cadenaConLaConversion final valdrá "110"
```

De esta manera no necesito invertir porque ya la tengo en el orden correcto. La idea es, en lugar de concatenar por la derecha, **concatenar** por la izquierda.

Ejercicio 6

> En una tienda de piezas de repuesto se realizan una seríe de ventas de las que tendremos que calcular el precio final a pagar. Escribe un programa teniendo en cuenta que los descuentos a efectuar están en función de:
>
> 1. La clase de comprador (clase A: 2%, clase B: 4% y clase C: 6%)
> 2. Del tipo de pieza (tipo 1: 8% y tipo 2: 10%).
>
> Crea un método **LeeDatos** que recoja el precio de la
> pieza, la clase de comprador y el tipo de pieza, y los devuelva mediante **tupla**.
> Escribe un método llamada **PrecioAPagar**, para calcular y devolver el precio final. A este método se le pasará la tupla y devolverá el valor del precio final, se hará a través de un switch C#8.
>
> 👁 **Nota:** Se termina la introducción de datos cuando un precio sea cero. El programa visualizara el precio parcial de cada venta y una vez terminadas estas, el importe total líquido obtenido en las ventas efectuadas (la suma de total de todas las ventas).

☑ Ejercicio 7

*El **calendario Gregoriano actual** obedece a la reforma del calendario juliano que ordenó el papa Gregorio XIII en 1582. Se decidió, después de algunas modificaciones, que en lo sucesivo fuesen bisiestos todos los años múltiplos de cuatro, pero que de los años seculares (los acabados en dos ceros) sólo fuesen bisiestos aquellos que fuesen múltiplos de cuatrocientos*.

En base a estos conceptos, deberemos construir un programa que dada una fecha (día, mes y año), nos devuelva como resultado el correspondiente día de la semana en texto (Lunes, Martes, Miércoles, etc.).

✋ **Importante:** La estructura del programa estará formada, además de por el método main, los métodos estáticos (Piensa cual sería el DEM lógico para estas definiciones):

- `(int dia, int mes, int año) LeeFecha()`
- `bool FechaValida(int dia, int mes, int año)`
- `bool Bisiesto(int año)`
- `string DiaSemana(int dia, int mes, int año)`

Consideraciones:

1. La entrada será válida si...
   - **año** → es mayor o igual que 1582.
   - **día** → cumple la condición de ser mayor que 1 y menor que 28, 29, 30 o 31 según corresponda.
   - **mes** → esté entre 1 y 12.
2. Antes de calcular el día de la semana, se tendrán que ajustar los datos de manera que: Si `mes <= 2`, tendremos que **sumarle 12 al mes** y **restarle uno al año**. El día de la semana se calculará con la siguiente operación:

```
diaSemana = (dia + 2*mes + 3*(mes + 1)/5 + año + año/4 - año/100 + año/400 + 2) % 7;
```

👁 **Nota:** Si diaSemana es 1 → Domingo, 2 → Lunes, 3 → Martes, … y 0 → Sábado..