

EJERCICIOS TEMA 5

(5.2.1) Crea una función llamada "BorrarPantalla", que borre la pantalla dibujando 25 líneas en blanco. No debe devolver ningún valor. Crea también un "Main" que permita probarla.

```
using System;

public class tema5
{
    public static void BorrarPantalla ()
    {
        Console.Write ("Escribo algo para que ahora se borre...");
        Console.ReadLine ();
        Console.Clear ();
    }
    public static void Main()
    {
        BorrarPantalla ();
    }
}
```

(5.2.2) Crea una función llamada "DibujarCuadrado3x3", que dibuje un cuadrado formato por 3 filas con 3 asteriscos cada una. Crea también un "Main" que permita probarla.

```
using System;

public class tema5
{
    public static void DibujarCuadrado3x3 ()
    {
        for (int i = 0; i < 3; i++) {
            for (int j=0; j<3; j++){
                Console.Write ('*');
            }
            Console.WriteLine ();
        }
    }
    public static void Main()
    {
        DibujarCuadrado3x3 ();
    }
}
```

(5.2.3) Descompón en funciones la base de datos de ficheros (ejemplo 04_06a), de modo que el "Main" sea breve y más legible (Pista: las variables que se compartan entre varias funciones deberán estar fuera de todas ellas, y deberán estar precedidas por la palabra "static").

```
using System;
```

```
public class ejemploCompleto
```

```
{  
    static fichero[] baseDatos=new fichero[5];  
    static int contadorFichas = 0;
```

```
    struct fichero
```

```
    {  
        public string nombre;  
        public long tamaño;  
    }
```

```
    public static int MostrarMenu(){
```

```
        Console.Write ("\n\t\tMENÚ DE PROGRAMA");  
        Console.Write ("\n\n1) Añadir datos de un nuevo fichero.");  
        Console.Write ("\n2) Mostrar los nombres de todos los ficheros almacenados.");  
        Console.Write ("\n3) Mostrar ficheros que sean de más de un cierto tamaño (por ejemplo, 2000 KB).");  
        Console.Write ("\n4) Ver todos los datos de un cierto fichero (a partir de su nombre).");  
        Console.Write ("\n5) Salir de la aplicación.");  
        Console.Write ("\n\nElija una opción del menú: ");
```

```
        return Convert.ToInt32 (Console.ReadLine ());
```

```
    }
```

```
    public static void AñadirFichero ()
```

```
    {  
        if (contadorFichas < 5) {  
            do {  
                Console.Write ("Introduzca el nombre del fichero: ");  
                baseDatos [contadorFichas].nombre = Console.ReadLine ();  
            } while (baseDatos [contadorFichas].nombre=="");  
            do {  
                Console.Write ("\nIntroduzca el tamaño del fichero: ");  
                baseDatos [contadorFichas].tamaño = Convert.ToInt64 (Console.ReadLine ());  
            } while (baseDatos [contadorFichas].tamaño<0);  
            Console.Write ("El tamaño tiene que ser positivo.");  
            while (baseDatos [contadorFichas].tamaño<0);  
            contadorFichas++;  
        } else {  
            Console.Write ("\nNo se pueden almacenar más. La base de datos está llena.\n")
```

```

;
    }
}

public static void MostrarNombres()
{
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].nombre != null) {
            Console.WriteLine ("-
En la {0}ª posición de la base de datos se encuentra el fichero: {1}", i+1, baseDatos [i].
nombre);
        }
    }
}

public static void MostrarFicheros()
{
    for (int i=0; i<baseDatos.Length; i++){
        if (baseDatos[i].tamaño>2000){
            Console.WriteLine ("-
En la {0}ª posición se encuentra el fichero \"{1}\" con el tamaño {2}.", i+1, baseDatos[
i].nombre, baseDatos[i].tamaño);
        }
    }
}

public static void VerDatos()
{
    string buscar;

    Console.Write ("Introduzca el fichero que quiere buscar: ");
    buscar=Console.ReadLine();
    for (int i = 0; i < baseDatos.Length; i++) {
        if (baseDatos [i].nombre == buscar) {
            Console.WriteLine ("-
En la {0}ª posición se encuentra el fichero \"{1}\" con el tamaño: {2}.", i+1, baseDatos
[i].nombre, baseDatos[i].tamaño);
        }
    }
}

public static void Eleccion(int num)
{
    switch (num) {
        case 1:
            AñadirFichero();
            break;
        case 2:
            MostrarNombres();
            break;
    }
}

```

```

        case 3:
            MostrarFicheros();
            break;
        case 4:
            VerDatos();
            break;
    }
}

public static void Main ()
{
    int num;

    do {
        num = MostrarMenu();

        Eleccion(num);

    } while (num != 5);
    Console.WriteLine ("\n\nEl programa ha finalizado.");
}
}

```

(5.3.1) Crea una función que dibuje en pantalla un cuadrado del ancho (y alto) que se indique como parámetro. Completa el programa con un Main que permita probarla.

```

using System;

public class tema5
{
    public static void DibujarCuadrado (int ancho)
    {
        for (int i = 0; i < ancho; i++) {
            for (int j=0; j<ancho; j++){
                Console.Write ('*');
            }
            Console.WriteLine ();
        }
    }
    public static void Main()
    {
        DibujarCuadrado (4);
    }
}

```

(5.3.2) Crea una función que dibuje en pantalla un rectángulo del ancho y alto que se indiquen como parámetros. Completa el programa con un Main que permita probarla.

```
using System;
```

```
public class tema5
```

```
{  
    public static void DibujarCuadrado (int ancho, int alto)  
    {  
        for (int i = 0; i < alto; i++) {  
            for (int j=0; j<ancho; j++){  
                Console.Write ('*');  
            }  
            Console.WriteLine ();  
        }  
    }  
    public static void Main()  
    {  
        DibujarCuadrado (4,5);  
    }  
}
```

(5.3.3) Crea una función que dibuje en pantalla un rectángulo hueco del ancho y alto que se indiquen como parámetros, formado por una letra que también se indique como parámetro. Completa el programa con un Main que pida esos datos al usuario y dibuje el rectángulo.

```
using System;
```

```
public class tema5
```

```
{
    public static void DibujarCuadrado (int ancho, int alto, char simbolo)
    {
        for (int i = 0; i < alto; i++) {
            for (int j=0; j<ancho; j++){
                if ((i == 0)|| (i==alto-1) || (j==0) || (j==ancho-1)){
                    Console.Write (simbolo);
                } else
                    Console.Write (' ');
            }
            Console.WriteLine ();
        }
    }
    public static void Main()
    {
        char simbolo;
        int alto, ancho;

        Console.Write ("Introduce un símbolo: ");
        simbolo = Convert.ToChar (Console.ReadLine ());
        Console.Write ("Introduce el alto del rectángulo: ");
        alto = Convert.ToInt32 (Console.ReadLine ());
        Console.Write ("Introduce el ancho del rectángulo: ");
        ancho = Convert.ToInt32 (Console.ReadLine ());

        DibujarCuadrado (ancho, alto, simbolo);
    }
}
```

(5.4.1) Crear una función que calcule el cubo de un número real (float) que se indique como parámetro. El resultado deberá ser otro número real. Probar esta función para calcular el cubo de 3.2 y el de 5.

```
using System;
```

```
public class tema5
```

```
{  
    public static float Cubo (float x)  
    {  
        return x*x*x;  
    }  
    public static void Main()  
    {  
        float num, resultado;  
  
        Console.Write ("Introduce un número: ");  
        num = float.Parse (Console.ReadLine ());  
  
        resultado=Cubo (num);  
  
        Console.Write ("El cubo de {0} es: {1}", num, resultado);  
    }  
}
```

(5.4.2) Crear una función que calcule el menor de dos números enteros que recibirá como parámetros. El resultado será otro número entero.

```
using System;
```

```
public class tema5
```

```
{  
    public static int Menor (int x, int y)  
    {  
        if (x < y)  
            return x;  
        else  
            return y;  
    }  
    public static void Main()  
    {  
        int num1, num2, resultado;  
  
        Console.Write ("Introduce un número: ");  
        num1 = int.Parse (Console.ReadLine ());  
        Console.Write ("Introduce otro número: ");  
        num2 = int.Parse (Console.ReadLine ());  
  
        resultado=Menor (num1, num2);  
  
        Console.Write ("El menor de los números introducidos es: {0}",resultado);  
    }  
}
```


(5.4.3) Crear una función llamada "Signo", que reciba un número real, y devuelva un número entero con el valor: -1 si el número es negativo, 1 si es positivo o 0 si es cero.

```
using System;
```

```
public class tema5
```

```
{  
    public static int Signo (float x)  
    {  
        if (x < 0)  
            return -1;  
        else {  
            if (x > 0)  
                return 1;  
            else  
                return 0;  
        }  
    }  
}  
public static void Main()  
{  
    float num;  
    int resultado;  
  
    Console.Write ("Introduce un número: ");  
    num = float.Parse (Console.ReadLine ());  
  
    resultado=Signo (num);  
  
    Console.Write ("El signo retornado es: {0}",resultado);  
}  
}
```

(5.4.4) Crear una función que devuelva la primera letra de una cadena de texto. Probar esta función para calcular la primera letra de la frase "Hola".

```
using System;
```

```
public class tema5
{
    public static string PrimeraLetra (string x)
    {
        return x.Substring (0, 1);
    }
    public static void Main()
    {
        string palabra;
        string resultado;

        Console.Write ("Introduce una palabra: ");
        palabra = Console.ReadLine ();

        resultado=PrimeraLetra (palabra);

        Console.Write ("La primera letra es: {0}",resultado);
    }
}
```

(5.4.5) Crear una función que devuelva la última letra de una cadena de texto. Probar esta función para calcular la última letra de la frase "Hola".

```
using System;
```

```
public class tema5
{
    public static string UltimaLetra (string x)
    {
        return x.Substring (x.Length-1, 1);
    }
    public static void Main()
    {
        string palabra;
        string resultado;

        Console.Write ("Introduce una palabra: ");
        palabra = Console.ReadLine ();

        resultado = UltimaLetra (palabra);

        Console.Write ("La última letra es: {0}",resultado);
    }
}
```

(5.4.6) Crear una función que reciba un número y calcule y muestre en pantalla el valor del perímetro y de la superficie de un cuadrado que tenga como lado el número que se ha indicado como parámetro.

```
using System;
```

```
public class tema5
```

```
{  
    public static void Cuadrado (int x)  
    {  
        int perimetro, superficie;  
        perimetro = x * 4;  
        superficie = x * x;  
        Console.WriteLine ("El perímetro es: {0}m y la superficie es: {1}m2",perimetro, super  
ficie);  
    }  
    public static void Main()  
    {  
        int lado;  
  
        Console.WriteLine ("Introduce el lado del cuadrado: ");  
        lado = int.Parse (Console.ReadLine ());  
  
        Cuadrado (lado);  
    }  
}
```

(5.5.1) Crear una función "pedirEntero", que reciba como parámetros el texto que se debe mostrar en pantalla, el valor mínimo aceptable y el valor máximo aceptable. Deberá pedir al usuario que introduzca el valor tantas veces como sea necesario, volvérselo a pedir en caso de error, y devolver un valor correcto. Probarlo con un programa que pida al usuario un año entre 1800 y 2100.

```
using System;
```

```
public class triangulo
```

```
{  
    public static int PedirEntero (string texto2, int nummin2, int nummax2 )  
    {  
        int año;  
  
        do{  
            Console.Write (texto2 +" entre {0} y {1}: ", nummin2, nummax2);  
            año = Convert.ToInt32 (Console.ReadLine());  
  
        } while ((año<nummin2)||año>nummax2));  
        return año;  
    }  
    public static void Main ()  
    {  
        string texto="Introduce un año";  
        int nummin=1800, nummax=2100, resultado;  
  
        resultado=PedirEntero (texto, nummin, nummax);  
  
        Console.Write ("El año {0} es válido.", resultado);  
    }  
}
```

(5.5.2) Crear una función "escribirTablaMultiplicar", que reciba como parámetro un número entero, y escriba la tabla de multiplicar de ese número (por ejemplo, para el 3 deberá llegar desde "3x0=0" hasta "3x10=30").

```
using System;
```

```
public class tablaMultiplicar
```

```
{  
    public static void EscribirTablaMultiplicar (int num2)  
    {  
        for (int i = 1; i <= 10; i++) {  
            Console.WriteLine ("{0} x {1} = {2}", num2, i, num2 * i);  
        }  
    }  
    public static void Main ()  
    {  
        int num;  
  
        Console.Write ("Escribe un número: ");  
        num = Convert.ToInt32 (Console.ReadLine ());  
  
        EscribirTablaMultiplicar (num);  
    }  
}
```

(5.5.3) Crear una función "esPrimo", que reciba un número y devuelva el valor booleano "true" si es un número primo o "false" en caso contrario.

```
using System;
```

```
public class numeroPrimo
```

```
{  
    public static bool EsPrimo (int num2)  
    {  
        bool primo=false;  
        int a = 0;  
  
        for (int i = 1; i <= num2; i++) {  
            if (num2 % i == 0) {  
                a++;  
            }  
            if (a == 2) {  
                primo = true;  
            } else  
                primo = false;  
        }  
  
        return primo;  
    }  
}
```

```
public static void Main ()
```

```
{  
    int num;  
  
    Console.Write ("Escribe un número: ");  
    num = Convert.ToInt32 (Console.ReadLine ());  
  
    if (EsPrimo(num))  
        Console.Write ("El número {0} es primo.",num);  
    else  
        Console.Write ("El número {0} NO es primo.",num);  
}  
}
```

(5.5.4) Crear una función que reciba una cadena y una letra, y devuelva la cantidad de veces que dicha letra aparece en la cadena. Por ejemplo, si la cadena es "Barcelona" y la letra es 'a', debería devolver 2 (porque la "a" aparece 2 veces).

```
using System;
```

```
public class cadena
```

```
{  
    public static int NumeroVecesLetra (string palabra2, char letra2)  
    {  
        int contador = 0;  
  
        for (int i = 0; i < palabra2.Length; i++) {  
            if (palabra2 [i] == letra2) {  
                contador++;  
            }  
        }  
        return contador;  
    }  
    public static void Main ()  
    {  
        string palabra;  
        char letra;  
  
        Console.Write ("Introduce una palabra: ");  
        palabra = Console.ReadLine ();  
        Console.Write ("Introduce una letra: ");  
        letra = Convert.ToChar (Console.ReadLine ());  
  
        Console.Write ("La letra \"{0}\" aparece {1} veces en la palabra \"{2}\".", letra, NumeroVecesLetra (palabra, letra), palabra);  
    }  
}
```

(5.5.5) Crear una función que reciba un numero cualquiera y que devuelva como resultado la suma de sus dígitos. Por ejemplo, si el número fuera 123 la suma sería 6.

```
using System;
```

```
public class MainClass
```

```
{  
    public static int SumaDigitos (int num2)  
    {  
        int suma = 0;  
        string cadena = num2.ToString ();  
  
        for (int i = 0; i < cadena.Length; i++) {  
            suma += Convert.ToInt32 (cadena.Substring (i,1));  
        }  
        return suma;  
    }  
    public static void Main ()  
    {  
        int num;  
  
        Console.Write ("Introduce un número: ");  
        num = Convert.ToInt32 (Console.ReadLine ());  
  
        Console.Write ("El resultado de la suma de los dígitos es: {0}", SumaDigitos (num  
));  
    }  
}
```


(5.5.6) Crear una función que reciba una letra y un número, y escriba un "triángulo" formado por esa letra, que tenga como anchura inicial la que se ha indicado. Por ejemplo, si la letra es * y la anchura es 4, debería escribir

```
****
```

```
***
```

```
**
```

```
*
```

```
using System;
```

```
public class dibujar
```

```
{
```

```
    public static void Triangulo (char letra2, int num2)
```

```
    {
```

```
        int ancho = num2;
```

```
        for (int i = 0; i < num2; i++) {
```

```
            for (int j = 0; j < ancho; j++) {
```

```
                Console.Write (letra2);
```

```
            }
```

```
            Console.WriteLine ();
```

```
            ancho--;
```

```
        }
```

```
    }
```

```
    public static void Main ()
```

```
    {
```

```
        char letra;
```

```
        int num;
```

```
        Console.Write ("Introduce una letra: ");
```

```
        letra = Convert.ToChar (Console.ReadLine ());
```

```
        Console.Write ("Introduce el ancho del triángulo: ");
```

```
        num = Convert.ToInt32 (Console.ReadLine ());
```

```
        Triangulo (letra, num);
```

```
    }
```

```
}
```

(5.7.1) Crear una función "Intercambiar", que intercambie el valor de los dos números enteros que se le indiquen como parámetro.

```
using System;
```

```
public class referencia
```

```
{  
    public static void Intercambia (ref int c, ref int d)  
    {  
        c++;  
        d++;  
    }  
    public static void Main ()  
    {  
        int a = 4, b = 8;  
  
        Console.WriteLine ("Los valores de a y b son: {0} y {1}", a, b);  
  
        Intercambia (ref a, ref b);  
  
        Console.WriteLine ("\n\nAhora los valores de a y b son: {0} y {1}", a, b);  
    }  
}
```

(5.7.2) Crear una función "Iniciales", que reciba una cadena como "Nacho Cabanes" y devuelva las letras N y C (primera letra, y letra situada tras el primer espacio), usando parámetros por referencia.

```
using System;
```

```
public class cadenas
```

```
{  
    public static void Iniciales (ref string cadena)  
    {  
        char delimitador = ' ';  
  
        string[] cadena2= cadena.Split (delimitador);  
  
        cadena = "";  
  
        for (int i = 0; i < cadena2.Length; i++) {  
            cadena = cadena + ' ' + cadena2 [i].Substring (0, 1);  
        }  
    }  
    public static void Main ()  
    {  
        string nombre = "Antonio Barba";  
  
        Console.Write ("La cadena es: {0}", nombre);  
  
        Iniciales (ref nombre);  
  
        Console.WriteLine ("\n\nAhora la cadena es: {0}", nombre);  
    }  
}
```

(5.9.1.2) Crear un programa que genere un número al azar entre 1 y 100. El usuario tendrá 6 oportunidades para acertarlo.

```
using System;
```

```
public class azar
```

```
{  
    public static void Main ()  
    {  
        int contador = 5, numeroUsuario, numeroAzar;  
  
        Random aleatorio = new Random ();  
  
        numeroAzar = aleatorio.Next (1, 101);  
  
        do {  
            Console.Write ("\nIntroduce un número entre 1 y 100: ");  
            numeroUsuario = Convert.ToInt32 (Console.ReadLine ());  
            if (contador == 0) {  
                Console.Write ("\nLo siento, has fallado. El número es: {0}", numeroAzar);  
                break;  
            } else {  
                if (numeroUsuario < numeroAzar) {  
                    Console.Write ("\nEl número que tienes que adivinar es mayor. Te quedan  
{0} oportunidades.", contador);  
                } else {  
                    if (numeroUsuario > numeroAzar) {  
                        Console.Write ("\nEl número que tienes que adivinar es menor. Te queda  
n {0} oportunidades.", contador);  
                    } else {  
                        Console.Write ("\nEnhorabuena. Has acertado!!!");  
                        break;  
                    }  
                }  
            }  
            contador--;  
        } while (contador >= 0);  
    }  
}
```

(5.9.1.2) Mejorar el programa del ahorcado (4.4.9.3), para que la palabra a adivinar no sea tecleado por un segundo usuario, sino que se escoja al azar de un "array" de palabras prefijadas (por ejemplo, nombres de ciudades).

```
using System;

public class ahorcado
{
    public static void Main ()
    {
        Random aleatorio = new Random ();

        string[] palabrasOcultas = { "bosque", "green peace", "la gata sobre el tejado de zin
c",
        "el universo", "universidad europea de madrid", "el gran gatsby" };
        int posicion = aleatorio.Next (0, palabrasOcultas.Length);
        string palabraElegida = palabrasOcultas [posicion];

        string palabraGuiones = "";

        for (int i = 0; i < palabraElegida.Length; i++) {
            if (palabraElegida [i] == ' ') {
                palabraGuiones += ' ';
            } else {
                palabraGuiones += '-';
            }
        }

        char letraUsuario;
        bool terminado = false;
        int oportunidades = 8;

        do {

            Console.WriteLine ("\nLa palabra oculta es: {0}", palabraGuiones);
            Console.WriteLine ("\nTienes {0} oportunidades.", oportunidades);
            Console.WriteLine ("\nIntroduce una letra: ");
            letraUsuario = Convert.ToChar (Console.ReadLine ());

            string palabraGuiones2="";

            for (int i = 0; i < palabraElegida.Length; i++) {
                if (palabraElegida[i]==letraUsuario){
                    palabraGuiones2+=letraUsuario;
                } else {
                    palabraGuiones2+=palabraGuiones[i];
                }
            }
            palabraGuiones=palabraGuiones2;
        }
```

```

        if (palabraGuiones.IndexOf('-')== -1){
            Console.WriteLine("\nEnhorabuena, has acertado. La palabra oculta es: {0}", palabraElegida);
            terminado=true;
        }

        if (palabraElegida.IndexOf(letraUsuario)== -1){
            oportunidades--;
        }
        if (oportunidades==0){
            Console.WriteLine("\nLo siento, has fallado. La palabra oculta es: {0}", palabraElegida);
            terminado=true;
        }

    } while (!terminado);
}
}

```

(5.9.1.5) Crea un programa que "dibuje" asteriscos en 100 posiciones al azar de la pantalla . Para ayudarte para escribir en cualquier coordenada, puedes usar un array de dos dimensiones (con tamaños 24 para el alto y 79 para el ancho), que primero rellenes y luego dibujes en pantalla.

```

using System;

public class asteriscos
{
    public static void Main ()
    {
        string [,] array = new string[24,79];

        RellenarArray (array);
        ImprimeArray (array);
    }
    public static string [,] RellenarArray (string [,] array2)
    {
        int cont = 0;

        Random aleatorio = new Random ();
        for (cont = 0; cont < 100; cont++) {
            int azar = aleatorio.Next (0, 24);
            int azar2 = aleatorio.Next (0, 79);
            array2 [azar, azar2] = "*";
        }
        return array2;
    }
    public static void ImprimeArray(string [,] array3)

```

```

{
    int filas = array3.GetLength (0);
    int columnas = array3.GetLength (1) ;

    for (int i = 0; i < filas; i++) {
        for (int j = 0; j < columnas; j++) {
            if (array3 [i, j] == null) {
                Console.Write (' ');
            } else {
                Console.Write (array3 [i, j]);
            }
        }
    }
}

```

Otra manera más sencilla de hacerlo:

```

using System;

public class dibujoAzar
{
    public static void Main ()
    {
        Random aleatorio = new Random();

        int contador = 0;

        for (int i = 0; i < 100; i++) {
            int azar = aleatorio.Next (0, 80);
            int azar2 = aleatorio.Next (0, 24);

            Console.SetCursorPosition (azar, azar2);
            Console.Write ('*');
            contador++;
        }
    }
}

```

(5.9.2.2) Crea un programa que halle cualquier raíz de un número. El usuario deberá indicar el número (por ejemplo, 2) y el índice de la raíz (por ejemplo, 3 para la raíz cúbica). Pista: hallar la raíz cúbica de 2 es lo mismo que elevar 2 a $1/3$.

```
using System;
```

```
public class raiz
```

```
{  
    public static void Main ()  
    {  
        int numero, indice;  
  
        Console.Write ("Introduce un número: ");  
        numero = Convert.ToInt32 (Console.ReadLine ());  
  
        Console.Write ("Introduce un índice de la raíz: ");  
        indice = Convert.ToInt32 (Console.ReadLine ());  
  
        Raiz (numero, indice);  
    }  
  
    public static void Raiz (double num, double indice2)  
    {  
        double resultado;  
  
        resultado = Math.Pow (num, (1 / indice2));  
  
        Console.Write ("El resultado de la raíz es: {0}", resultado);  
    }  
}
```


(5.9.2.3) Haz un programa que resuelva ecuaciones de segundo grado, del tipo $ax^2 + bx + c = 0$. El usuario deberá introducir los valores de a, b y c. Se deberá crear una función "raicesSegundoGrado", que recibirá como parámetros los coeficientes a, b y c, así como las soluciones x1 y x2 (por referencia). Deberá devolver los valores de las dos soluciones x1 y x2. Si alguna solución no existe, se devolverá como valor 100.000 para esa solución. Pista: la solución se calcula con $x = \frac{-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}}{2 \cdot a}$

```
using System;
```

```
public class ecuacion
```

```
{
    public static void Main ()
    {
        double a1, b1, c1, x1=0, x2=0;

        Console.Write ("Introduce el valor de a: ");
        a1 = Convert.ToInt32 (Console.ReadLine ());

        Console.Write ("Introduce el valor de b: ");
        b1 = Convert.ToInt32 (Console.ReadLine ());

        Console.Write ("Introduce el valor de c: ");
        c1 = Convert.ToInt32 (Console.ReadLine ());

        EcuacionSegundoGrado (a1,b1,c1, ref x1, ref x2);

        Console.Write ("Los resultados de la ecuación son: x1={0} y x2={1}", x1, x2);
    }

    public static void EcuacionSegundoGrado (double a, double b, double c, ref double x1, ref double x2)
    {
        double raiz;

        raiz = Math.Sqrt ((b * b) - (4 * a * c));

        x1 = ((-b + raiz) / (2 * a));
        if (double.IsInfinity (x1) || double.IsNaN (x1)) {
            x1 = 100000;
        }

        x2 = ((-b - raiz) / (2 * a));
        if (double.IsInfinity (x2) || double.IsNaN (x2)) {
            x2 = 100000;
        }
    }
}
```

(5.9.2.5) Haz un programa que pida al usuario 5 datos numéricos, los guarde en un array, pida un nuevo dato y muestre el valor del array que se encuentra más cerca de ese dato en valor absoluto (es decir, el más próximo, sea mayor que él o menor que él).

```
using System;
```

```
public class ecuacion
```

```
{  
    public static void Main ()  
    {  
        int[] array = new int[5];  
        int nuevo;  
  
        for (int i = 0; i < array.Length; i++) {  
            Console.Write ("Introduce el {0}º número: ", i + 1);  
            array [i] = Convert.ToInt32 (Console.ReadLine ());  
        }  
  
        Console.Write ("Introduce un nuevo número: ");  
        nuevo = Convert.ToInt32 (Console.ReadLine ());  
  
        Console.Write ("El valor del array más próximo a {0} en su valor absoluto es {1}."  
        , nuevo, mayorMasCercanoAbsoluto (array, nuevo));  
    }  
    public static int mayorMasCercanoAbsoluto(int[] array2, int numeroComparar)  
    {  
        int elegido = array2 [0];  
        int comparacionAbsoluta = int.MaxValue;  
  
        for (int i = 0; i < array2.Length; i++) {  
            int valueAbsoluto = Math.Abs (array2 [i]);  
            int aux = Math.Abs(valueAbsoluto - numeroComparar);  
            if (aux < comparacionAbsoluta) {  
                comparacionAbsoluta = aux;  
                elegido = array2 [i];  
            }  
        }  
        return elegido;  
    }  
}
```

RECURSIVIDAD

Ejemplo:

```
using System;
public class Recursividad
{
    public static long fact(int n)
    {
        long factorial;
        if (n == 1) {
            return 1;
        } else {
            factorial = n * fact (n - 1);
            return factorial;
        }
    }
    public static void Main()
    {
        int num;
        Console.Write ("Introduzca un número entero: ");
        num = Convert.ToInt32(System.Console.ReadLine());
        Console.WriteLine("El factorial de {0} es: {1}", num, fact(num));
    }
}
```

Caso base (1ª condición de la recursiva).

Hasta que no se cumple el caso base no termina la recursividad, es decir, $n=1$. No devuelve nada hasta que no acaba la pila de llamadas recursivas. El proceso que sigue la función es:

fact = 3 * factorial(2)

fact = 2 * factorial(1)

fact = 1

fact = 2 * 1

fact = 3 * 2

(5.10.1) Crear una función que calcule el valor de elevar un número entero a otro número entero (por ejemplo, 5 elevado a 3 = $5^3 = 5 \cdot 5 \cdot 5 = 125$). Esta función se debe crear de forma recursiva. Piensa cuál será el caso base (qué potencia se puede calcular de forma trivial) y cómo pasar del caso “n-1” al caso “n” (por ejemplo, si sabes el valor de 5^4 , cómo hallarías el de 5^5 a partir de él).

```
using System;
```

```
public class recursividad
```

```
{  
    public static void Main ()  
    {  
        int bas, exp;  
        Console.Write ("Introduce la base: ");  
        bas = Convert.ToInt32 (Console.ReadLine ());  
        Console.Write ("Introduce el exponente: ");  
        exp = Convert.ToInt32 (Console.ReadLine ());  
  
        Console.Write ("La potencia de {0} elevado a {1} es: {2}", bas, exp, Potencia (bas  
, exp));  
    }  
  
    public static int Potencia (int bas2, int exp2)  
    {  
        if (exp2==0) {  
            return 1;  
        } else {  
            return bas2 * Potencia (bas2, exp2-1);  
        }  
    }  
}
```

(5.10.2) Como alternativa, crear una función que calcule el valor de elevar un número entero a otro número entero de forma NO recursiva (lo que llamaremos "de forma iterativa"), usando la orden "for".

```
using System;
```

```
public class recursividad
```

```
{  
    public static void Main ()  
    {  
        int bas, exp;  
        Console.Write ("Introduce la base: ");  
        bas = Convert.ToInt32 (Console.ReadLine ());  
        Console.Write ("Introduce el exponente: ");  
        exp = Convert.ToInt32 (Console.ReadLine ());  
  
        Console.Write ("La potencia de {0} elevado a {1} es: {2}", bas, exp, Potencia (bas  
, exp));  
    }  
  
    public static int Potencia (int bas2, int exp2)  
    {  
        int resultado = 1;  
  
        for (int i = 1; i <= exp2; i++) {  
            resultado *= bas2;  
        }  
        return resultado;  
    }  
}
```

(5.10.3) Crear un programa que emplee recursividad para calcular un número de la serie Fibonacci (en la que los dos primeros elementos valen 1, y para los restantes, cada elemento es la suma de los dos anteriores).

```
using System;
```

```
public class recursividad
```

```
{  
    public static void Main ()  
    {  
        int posicion;  
        Console.WriteLine ("¿Qué posición de la serie de Fibonacci quieres ver: ");  
        posicion = Convert.ToInt32 (Console.ReadLine ());  
  
        Console.WriteLine ("El {0}º número en la serie de Fibonacci es: {1}", posicion, Fibonacc  
        i (posicion));  
    }  
  
    public static long Fibonacci(long posicion)  
    {  
        long resultado;  
  
        if (posicion < 3) {  
            return 1;  
        } else {  
            resultado = Fibonacci (posicion - 1) + Fibonacci (posicion - 2);  
            return resultado;  
        }  
    }  
}
```

Hecho de forma iterativa

```
using System;
```

```
public class recursividad
```

```
{  
    public static void Main ()  
    {  
        int posicion;  
        Console.WriteLine ("¿Qué posición de la serie de Fibonacci quieres ver: ");  
        posicion = Convert.ToInt32 (Console.ReadLine ());  
  
        Console.WriteLine ("El {0}º número en la serie de Fibonacci es: {1}", posicion, Fibona  
        cci (posicion));  
    }  
  
    public static int Fibonacci (int posicion2)  
    {  
        int resultado = 0;
```

```

int resultado2 = 1;

if (posicion2 == 1) {
    return 1;
} else {
    for (int i = 1; i <= posicion2; i++) {
        int temp = resultado;
        resultado = resultado2;
        resultado2 = temp + resultado2;
    }
}
return resultado;
}
}

```

(5.10.6) Crear un programa que emplee recursividad para dar la vuelta a una cadena de caracteres (por ejemplo, a partir de "Hola" devolvería "aloH"). La función recursiva se llamará "Invertir(cadena)". Como siempre, analiza cuál será el caso base (qué longitud debería tener una cadena para que sea trivial darle la vuelta) y cómo pasar del caso "n-1" al caso "n" (por ejemplo, si ya has invertido las 5 primeras letras, que ocurriría con la de la sexta posición).

```
using System;
```

```
public class recursividad
```

```

{
    public static void Main ()
    {
        string palabra;
        Console.WriteLine ("Escribe una palabra: ");
        palabra = Console.ReadLine ();

        Console.WriteLine ("La palabra \"{0}\" al revés es: {1}", palabra, Reves (palabra));
    }

    public static string Reves (string palabra2)
    {
        string subcadena;

        if (palabra2.Length == 1) {
            return palabra2;
        } else {
            subcadena = palabra2.Substring (palabra2.Length - 1, 1);
        }
        return subcadena + Reves (palabra2.Remove (palabra2.Length-1, 1));
    }
}

```

(5.10.7) Crear, tanto de forma recursiva como de forma iterativa, una función diga si una cadena de caracteres es simétrica (un palíndromo). Por ejemplo, "DABALEARROZALAZORRAELABAD" es un palíndromo.

```
using System;
```

```
public class recursividad
```

```
{
```

```
    public static void Main ()
```

```
    {
```

```
        string palabra;
```

```
        Console.Write ("Escribe una frase: ");
```

```
        palabra = Console.ReadLine ();
```

```
        Console.Write ("¿La palabra \"{0}\" es un palíndromo? {1}", palabra, EsPalindromo (palabra));
```

```
    }
```

```
    public static bool EsPalindromo(string frase)
```

```
    {
```

```
        if (frase.Length <= 1)
```

```
            return true;
```

```
        else
```

```
        {
```

```
            if ( frase[0] != frase[ frase.Length - 1 ] )
```

```
                return false;
```

```
            else
```

```
                return EsPalindromo( frase.Substring( 1, frase.Length-2 ) );
```

```
        }
```

```
    }
```

```
}
```


(5.11.1) Crear un programa llamado "suma", que calcule (y muestre) la suma de dos números que se le indiquen como parámetro. Por ejemplo, si se teclea "suma 2 3" deberá responder "5", y si se teclea "suma 2" deberá responder "no hay suficientes datos y devolver un código de error 1.

```
using System;
```

```
public class tema5
```

```
{  
    public static void Main(string[] args) {  
  
        int num1=0, num2=0;  
        string operacion="";  
  
        if ((args.Length < 3) || (args.Length>3)){  
            Console.WriteLine ("No hay suficientes datos o son demasiados.");  
            Environment.Exit (1);  
        } else {  
            num1 = Convert.ToInt32 (args [1]);  
            num2 = Convert.ToInt32 (args [2]);  
            operacion=args[0];  
        }  
        if (operacion == "suma") {  
            suma (num1, num2);  
        } else {  
            if (operacion == "multiplicacion") {  
                multiplicacion (num1, num2);  
            } else {  
                Console.Write ("La operación no es correcta.");  
            }  
        }  
        //return 0;  
    }  
  
    public static void suma (int numero1, int numero2)  
    {  
        Console.WriteLine ("La suma de " + numero1 + " y " + numero2 + " es: " + (num  
ero1 + numero2));  
    }  
    public static void multiplicacion (int numero1, int numero2)  
    {  
        Console.WriteLine ("La multiplicación de " + numero1 + " y " + numero2 + " es:  
" + (numero1 * numero2));  
    }  
}
```