

Ejemplo 2

En este ejemplo solo se muestra que un arreglo de caracteres en C# se puede crear de tres maneras distintas: directamente con llaves, usando `new char[]` o declarando el arreglo vacío y llenando cada posición. El punto es ver que todas producen un arreglo válido y que en los tres casos el tamaño se obtiene igual con `.Length`.

```
char[] a1 = { 'a', 'b', 'c' };  
char[] a2 = new[] { 'a', 'b', 'c' };  
char[] a3 = new char[3] { 'a', 'b', 'c' };
```

```
MostrarTamaño("a1", a1);  
MostrarTamaño("a2", a2);  
MostrarTamaño("a3", a3);
```

Este código crea tres arreglos de caracteres usando diferentes formas. Luego muestra cuántos elementos tiene cada arreglo utilizando una función que imprime su tamaño. Al final espera una tecla para cerrar.

Codigo

```
using System;

namespace Ejemplo2
{
    class Program
    {
        static void Main(string[] args)
        {
            char[] a1 = { 'a', 'b', 'c' };
            char[] a2 = new[] { 'a', 'b', 'c' };
            char[] a3 = new char[3] { 'a', 'b', 'c' };

            MostrarTamaño("a1", a1);
            MostrarTamaño("a2", a2);
            MostrarTamaño("a3", a3);

            Console.ReadKey();
        }

        static void MostrarTamaño(string nombre, char[] arreglo)
        {
            Console.WriteLine($"Tamaño {nombre}[]: {arreglo.Length}");
        }
    }
}
```

Ejemplo 3

el programa pide un mensaje y lo convierte en un arreglo de caracteres usando ToCharArray(). Después recorre ese arreglo para imprimir cada carácter. Básicamente sirve para entender que un texto se puede descomponer letra por letra y manejarlo como un array.

```
Console.Write("* Ingresa mensaje: ");  
char[] c = Console.ReadLine().ToCharArray();
```

```
Console.Write("c[{0}] = ", c.Length);  
for (int i = 0; i < c.Length; i++)  
{  
    Console.Write("[{0}]", c[i]);  
}
```

Este código pide un mensaje al usuario, lo convierte en un arreglo de caracteres y luego muestra cada carácter uno por uno.

Codigo

```
using System;

namespace Ejemplo3
{
    class Program
    {
        static void Main(string[] args)
        {

            char[] c;
            Console.Write("* Ingresa mensaje: ");
            c = Console.ReadLine().ToCharArray();

            Console.WriteLine();
            Console.WriteLine("* Mensaje ingresado.");

            Console.Write("c[{0}] = ", c.Length);
            for (int i = 0; i < c.Length; i++)
            {
                Console.Write("[{0}]", c[i]);
            }

            Console.ReadKey();
        }
    }
}
```

Ejemplo 4

En este ejercicio se juntan dos arreglos: uno que contiene la palabra “Bienvenido ” y otro que contiene el nombre ingresado por el usuario. Lo que hace el programa es ampliar el primer arreglo y copiar ahí cada carácter del nombre. Es una forma manual de concatenar dos textos usando arreglos.

El código toma la palabra ‘Bienvenido ’, le agrega el nombre que escribe el usuario y muestra la frase completa.

Ingresar tu nombre : Ambmtuk

frase: Bienvenido Ambtmuk

Codigo

```
using System;

namespace Ejemplo4
{
    class Program
    {
        static void Main(string[] args)
        {
            char[] saludo = "Bienvenido ".ToCharArray();

            Console.Write("* Ingresar su nombre: ");
            char[] nombre = Console.ReadLine().ToCharArray();

            Array.Resize(ref saludo, saludo.Length + nombre.Length);

            for (int i = 0; i < nombre.Length; i++)
                saludo[saludo.Length - nombre.Length + i] = nombre[i];

            Console.WriteLine();
            Console.Write("* Frase: ");

            foreach (char c in saludo)
                Console.Write(c);

            Console.ReadKey();
        }
    }
}
```

Ejemplo 5

Este ejemplo recibe una frase y luego cuenta cuántas vocales tiene. Convierte cada letra a mayúscula y usa un switch para sumar a cada contador. La idea es practicar cómo recorrer un arreglo e identificar elementos específicos dentro de él.



**El programa pide al usuario que escriba una frase.
Luego recorre cada carácter y convierte cada uno a mayúscula para compararlo.
Con un switch, verifica si la letra es una vocal (A, E, I, O, U) y suma 1 al contador correspondiente.
Al final, muestra cuántas veces aparece cada vocal en la frase.**

Codigo

```
using System;

namespace Ejemplo5
{
    class Program
    {
        static void Main(string[] args)
        {
            char[] frase;
            int lim, cont_a = 0, cont_e = 0,
            cont_i = 0, cont_o = 0, cont_u = 0;

            Console.Write("Digite una frase: ");
            frase = Console.ReadLine().ToCharArray();
            lim = frase.Length;
```

```
for (int i = 0; i< lim; i++)
{
    switch (char.ToUpper(frase[i]))
    {
        case 'A':
            cont_a++;
            break;
        case 'E':
            cont_e++;
            break;

        case 'I':
            cont_i++;
            break;
        case 'O':
            cont_o++;
            break;
        case 'U':
            cont_u++;
            break;
    }
}

Console.WriteLine("\nContador de vocales.");
Console.WriteLine("* Vocal a: {0}", cont_a);
Console.WriteLine("* Vocal e: {0}", cont_e);
Console.WriteLine("* Vocal i: {0}", cont_i);
Console.WriteLine("* Vocal o: {0}", cont_o);
Console.WriteLine("* Vocal u: {0}", cont_u);

Console.ReadKey();
}
}
```

Ejemplo 6

Se ingresa un solo carácter y el programa revisa su código ASCII para saber si es número, letra o símbolo. Si está dentro del rango de números, dice que es número; si está en el rango de letras, dice que es letra; si no, lo toma como símbolo. Es un ejercicio para reconocer tipos de caracteres usando ASCII.

El programa pide un carácter al usuario.

Convierte ese carácter a mayúscula y obtiene su código ASCII.

Luego revisa:

- Si el ASCII está entre 48 y 57, es un número.
- Si está entre 65 y 90, es una letra.
- Si no está en esos rangos, entonces es un símbolo.

Al final muestra el carácter digitado y qué tipo de carácter es.

Digite un caracter: @
Resultado: @, es un simbolo

Codigo

```
using System;

namespace Ejemplo6
{
    class Program
    {
        static void Main(string[] args)
        {

            string r;
            char c;
            int ascii;

            Console.Write("Digite un caracter: ");
            c = char.Parse(Console.ReadLine());

            ascii = (int)char.ToUpper(c);

            if (ascii >= 48 && ascii <= 57)
            {
                r = "es un numero";
            }
            else if (ascii >= 65 && ascii <= 90)
            {
                r = "es una letra";
            }
            else
            {
                r = "es una simbolo";
            }
            Console.WriteLine();
            Console.WriteLine("* Resultado: {0}, {1}", c, r);
            Console.ReadKey();
        }
    }
}
```

Ejemplo 7

Este ejercicio tiene tres funciones: una para ingresar los datos al arreglo, otra para invertir los elementos y otra para imprimirlos. Al final se ve cómo un arreglo puede pasarse entre funciones y modificarse. Es básicamente práctica de modularización y manejo de arreglos enteros.

El programa lee un arreglo, lo invierte cambiando los valores de lugar y luego lo imprime en pantalla

```
Ingresa tamaño del array: 5
```

```
Ingresa datos al array:
```

```
a[0] = 1
```

```
a[1] = 2
```

```
a[2] = 3
```

```
a[3] = 4
```

```
a[4] = 5
```

```
Impresión de datos del array:
```

```
a[5] = [1][2][3][4][5]
```

Codigo

```
using System;
```

```
namespace Ejemplo7
```

```
{
    class Program
    {
        static void imprimir(int[] a)
        {
            int size = a.Length;
            Console.Write("a[{0}] = ", size);
            for (int i = 0; i < size; i++)
            {
                Console.Write("[{0}]", a[i]);
            }
        }

        static void invertir(int[] a)
        {
            int aux;
            int size = a.Length;
            for (int i = 0; i < size / 2; i++)
            {
                aux = a[i];
                a[i] = a[size - i - 1];
                a[size - i - 1] = aux;
            }
        }
    }
}
```

```
static int[] ingreso()
```

```
{
    int[] a; int size;
    Console.Write("Ingresar tamaño del array: ");
    size = int.Parse(Console.ReadLine());
    a = new int[size];
```

```

    Console.WriteLine("\nIngresar datos al array:");
    for (int i = 0; i < size; i++)
    {
        Console.Write("a[{0}] = ", i);
        a[i] = int.Parse(Console.ReadLine());
    }
    return a;
}
```

```
static void Main(string[] args)
```

```
{
    int[] a;
    a = ingreso();
    invertir(a);
    Console.WriteLine("\nImpresión de datos del array:");
    imprimir(a);
    Console.ReadKey();
}
}
```

Ejemplo 8

Es parecido al anterior, pero con caracteres. Se ingresa una frase, se convierte en arreglo, luego se invierte cambiando la primera letra por la última, la segunda por la penúltima, etc. Finalmente lo imprime. El objetivo es mostrar cómo invertir un texto usando un arreglo.

```
Digite una frase: Software
Frase invertida:
c[8] = [e][r][a][w][t][f][o][S]
```

El programa pide una frase, la convierte en un arreglo, le da la vuelta intercambiando letras de adelante con las de atrás y luego muestra la frase invertida.

Codigo

```
using System;

namespace Ejemplo8
{
    class Program
    {
        static void invertir(char[] c)
        {
            char aux;
            int size = c.Length;
            for (int i = 0; i < size / 2; i++)
            {
                aux = c[i];
                c[i] = c[size - i - 1];
                c[size - i - 1] = aux;
            }
        }

        static void imprimir(char[] c)
        {
            int size = c.Length;
            Console.Write("c[{0}] = ",size);
            for (int i = 0; i< size; i++)
            {
                Console.Write("[{0}]",c[i]);
            }
        }

        static char[] ingreso()
        {
            char[] c;
            Console.Write("Digite una frase: ");
            c = Console.ReadLine().ToCharArray();
        }
    }
}
```

```
        return c;
    }

    static void Main(string[] args)
    {
        char[] c;

        c = ingreso();
        invertir(c);
        Console.WriteLine("\nFrase invertida: ");
        imprimir(c);

        Console.ReadKey();
    }
}
```