18. Realice una función, que lea un número de máximo 8 cifras y luego sume cada dígito hasta obtener un valor de un solo dígito.

using System;

namespace ConsoleApp8

{

class Program

{

static void Main()

{

Console.Write("Ingrese un número de hasta 8 cifras: ");

string \_input = Console.ReadLine();

if (IsValid(\_input))

{

int \_numero = int.Parse(\_input);

int \_resultado = AddNumbers(\_numero);

Console.WriteLine("La suma de los dígitos hasta obtener un solo dígito es: " + \_resultado);

Console.ReadLine();

}

else

{

Console.WriteLine("Entrada no válida. Asegúrese de ingresar un número de hasta 8 cifras.");

Console.ReadLine();

}

}

static bool IsValid(string input)

{

// comprueba si es nulo, menor a 8 y que el texto se pueda convertir a int

return !string.IsNullOrEmpty(input) && input.Length <= 8 && int.TryParse(input, out \_);

}

static int AddNumbers(int numero)

{

//revisa que tenga mas de un digito

while (numero >= 10)

{

int suma = 0;

// suma los digitos

while (numero > 0)

{

suma += numero % 10;

//eliminamos el segundo digito ya que lo estamos procesando uno a la vez

numero /= 10;

}

numero = suma;

}

return numero;

}

}

}

19. Realice una función que invierta un array de números enteros

using System;

class Program

{

static void Main()

{

int[] \_numeros = { 1, 2, 3, 4, 5 };

Console.WriteLine("Array original:");

PrintArray(\_numeros);

InverseArray(\_numeros);

Console.WriteLine("Array invertido:");

PrintArray(\_numeros);

Console.ReadLine();

}

static void InverseArray(int[] array)

{

int \_Lenght = array.Length;

int \_Start = 0;

int \_End = \_Lenght - 1;

while (\_Start < \_End)

{

// Intercambiar elementos desde los extremos hacia el centro

int temp = array[\_Start];

array[\_Start] = array[\_End];

array[\_End] = temp;

// Mueve los índices hacia el centro

\_Start++;

\_End--;

}

}

static void PrintArray(int[] array)

{

foreach (int numero in array)

{

Console.Write(numero + " ");

}

Console.WriteLine();

}

}

7. Lea una cadena de números enteros positivos y luego cree un array con los números de la cadena, se debe validar que la cadena contenga números...

using System;

class Program

{

static void Main()

{

Console.Write("Ingrese una cadena de números enteros positivos separados por espacios: ");

string \_input = Console.ReadLine();

if (ValidateNumbers(\_input))

{

int[] \_numeros = CreateArray(\_input);

Console.WriteLine("Array creado:");

PrintArray(\_numeros);

}

else

{

Console.WriteLine("La cadena no contiene números enteros positivos válidos.");

}

Console.ReadLine();

}

static bool ValidateNumbers(string input)

{

// Verifica si la cadena contiene solo números enteros positivos

foreach (char caracter in input)

{

if (!char.IsDigit(caracter) && caracter != ' ')

{

return false;

}

}

return true;

}

static int[] CreateArray(string input)

{

// Dividimos} la cadena en segmentos usando espacios como delimitadores, creamos el array, convertimos cada segmento en un Int y lo almacenamos

string[] segmentos = input.Split(' ');

int[] numeros = new int[segmentos.Length];

for (int i = 0; i < segmentos.Length; i++)

{

if (int.TryParse(segmentos[i], out int numero))

{

numeros[i] = numero;

}

else

{

// Manejamos el caso en el que un segmento no sea un número entero válido

Console.WriteLine("Error al convertir " + segmentos[i] + " a un número entero.");

}

}

return numeros;

}

static void PrintArray(int[] array)

{

foreach (int numero in array)

{

Console.Write(numero + " ");

}

Console.WriteLine();

}

}

12. Dada una matriz nxm de caracteres, elimine las vocales de la matriz y reemplácelas con una x.

using System;

class Program

{

static void Main()

{

char[,] \_matriz = {

{'p', 'r', 'U', 'E', 'a'},

{'h', 'e', 'l', 'l', 'o'},

{'t', 'a', 'A', 's', 'i'},

{'i', 's', 'G', 't', 'e'}

};

Console.WriteLine("Matriz original:");

PrintMatrix(\_matriz);

ReplaceVocals(\_matriz);

Console.WriteLine("\nMatriz modificada:");

PrintMatrix(\_matriz);

Console.ReadLine();

}

static void ReplaceVocals(char[,] matriz)

{

int filas = matriz.GetLength(0);

int columnas = matriz.GetLength(1);

// recorre la matriz y reemplazar las vocales con x

for (int i = 0; i < filas; i++)

{

for (int j = 0; j < columnas; j++)

{

char caracter = matriz[i, j];

if (EsVocal(caracter))

{

matriz[i, j] = 'x';

}

}

}

}

static bool EsVocal(char c)

{ //comprueba si es vocal

switch (c)

{

case 'a':

return true;

case 'A':

return true;

case 'e':

return true;

case 'E':

return true;

case 'i':

return true;

case 'I':

return true;

case 'o':

return true;

case 'O':

return true;

case 'u':

return true;

case 'U':

return true;

}

return false;

}

static void PrintMatrix(char[,] matriz)

{

// imprime las matriceds

int filas = matriz.GetLength(0);

int columnas = matriz.GetLength(1);

for (int i = 0; i < filas; i++)

{

for (int j = 0; j < columnas; j++)

{

Console.Write(matriz[i, j] + " ");

}

Console.WriteLine();

}

}

}

26. Usando un ciclo while, ingrese n valores flotantes, cada que ingrese un número acumulelo en una variable, al final calcule el promedio de los números ingresados.

using System;

class Program

{

static void Main()

{

Console.Write("Ingrese la cantidad de numeros que quiere sumar y sacar el promedio: ");

int n = int.Parse(Console.ReadLine());

float suma = 0;

int contador = 0;

while (contador < n)

{

int posicion = (contador + 1);

Console.Write("Ingrese el " + posicion + " numero: ");

float numero = float.Parse(Console.ReadLine());

suma += numero;

contador++;

}

// calculamos el promedio

float promedio;

if (n > 0)

{

promedio = suma / n;

}

else

{

promedio = 0;

}

Console.WriteLine("La suma de los" + n + " números es: " + suma);

Console.WriteLine("El promedio de los " + n + " números es: " + promedio);

Console.ReadLine();

}

}