Lab 07

Q1.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Q1

{

static class ArrayProcessor

{

public static int[] ProcessArray(int size)

{

int[] arr = new int[size \* 2];

int index = 0;

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

{

Console.Write($"Enter value {i + 1}: ");

if (int.TryParse(Console.ReadLine(), out int value))

{

arr[index++] = value;

arr[index++] = 0;

}

else

{

Console.WriteLine("Invalid input! Please enter a valid integer value.");

return null;

}

}

return arr;

}

}

class Program

{

static void Main(string[] args)

{

Console.Write("Enter the size of the array: ");

if (int.TryParse(Console.ReadLine(), out int size) && size > 0)

{

int[] processedArray = ArrayProcessor.ProcessArray(size);

if (processedArray != null)

{

Console.WriteLine("Processed array values:");

foreach (int value in processedArray)

{

Console.Write($"{value} ");

}

}

}

else

{

Console.WriteLine("Invalid input! Please enter a valid positive integer value for the array size.");

}

Console.ReadKey();

}

}

}

Q2.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Q26

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter the size of the arrays: ");

if (int.TryParse(Console.ReadLine(), out int size) && size > 0)

{

int[] array1 = new int[size];

int[] array2 = new int[size];

int[] resultArray = new int[size];

Console.WriteLine("Enter values for the first array:");

InputArrayValues(array1);

Console.WriteLine("Enter values for the second array:");

InputArrayValues(array2);

int scalarSum = ScalarSum(array1) + ScalarSum(array2);

Console.WriteLine($"Scalar Sum: {scalarSum}");

VectorSum(array1, array2, resultArray);

Console.WriteLine("Vector Sum:");

PrintArray(resultArray);

VectorProduct(array1, array2, resultArray);

Console.WriteLine("Vector Product:");

PrintArray(resultArray);

int scalarProduct = ScalarProduct(array1, array2);

Console.WriteLine($"Scalar Product: {scalarProduct}");

}

else

{

Console.WriteLine("Invalid input! Please enter a valid positive integer value for the array size.");

}

Console.ReadKey();

}

static void InputArrayValues(int[] arr)

{

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

{

Console.Write($"Enter value {i + 1}: ");

if (int.TryParse(Console.ReadLine(), out int value))

{

arr[i] = value;

}

else

{

Console.WriteLine("Invalid input! Please enter a valid integer value.");

return;

}

}

}

static int ScalarSum(int[] arr)

{

int sum = 0;

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

{

sum += arr[i];

}

return sum;

}

static void VectorSum(int[] arr1, int[] arr2, int[] resultArr)

{

for (int i = 0; i < arr1.Length; i++)

{

resultArr[i] = arr1[i] + arr2[i];

}

}

static void VectorProduct(int[] arr1, int[] arr2, int[] resultArr)

{

for (int i = 0; i < arr1.Length; i++)

{

resultArr[i] = arr1[i] \* arr2[i];

}

}

static int ScalarProduct(int[] arr1, int[] arr2)

{

int sum = 0;

for (int i = 0; i < arr1.Length; i++)

{

sum += arr1[i] \* arr2[i];

}

return sum;

}

static void PrintArray(int[] arr)

{

foreach (int value in arr)

{

Console.Write($"{value} ");

}

Console.WriteLine();

}

}

}