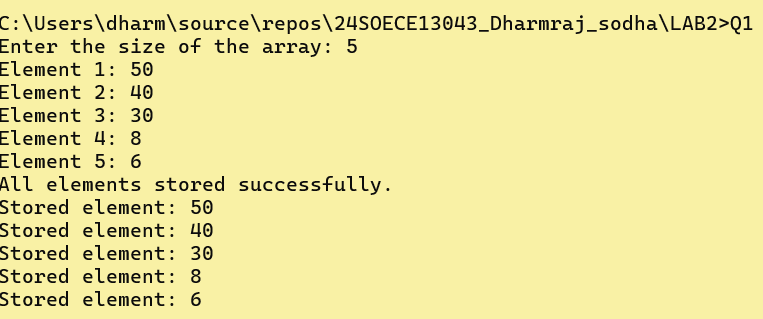
**Tutorial – 2**

**1. Write a C# Sharp program that stores elements in an array and prints them.**

| **using System;  class Q1 {  static void Main()  {  //stoeres elements in an array and prints them  int size;  Console.Write("Enter the size of the array: ");  try  {  size = Convert.ToInt32(Console.ReadLine());  if (size <= 0)  {  Console.WriteLine("Size must be a positive integer. Defaulting to 10.");  size = 10;  }  }  catch (FormatException)  {  Console.WriteLine("Invalid input! Defaulting size to 10.");  size = 10;  }  int[] arr = new int[size];  for (int i = 0; i < size; i++) {  Console.Write($"Element {i + 1}: ");  try  {  arr[i] = Convert.ToInt32(Console.ReadLine());  }  catch (FormatException)  {  Console.WriteLine("Invalid input! Please enter a valid integer.");  i--; // Decrement i to repeat this iteration  }  }  Console.WriteLine("All elements stored successfully.");  foreach (int element in arr)  {  Console.WriteLine($"Stored element: {element}");  }  }  }** |
| --- |

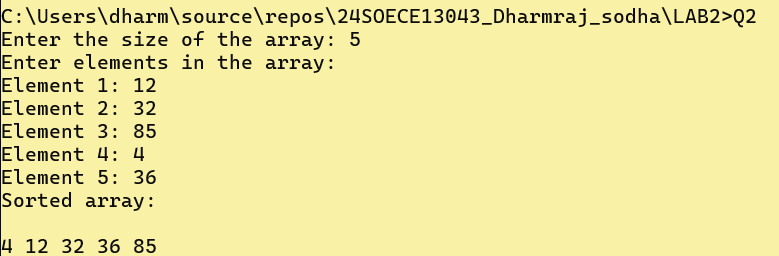
**Output:**

****

**2. Write a program of sorting an array. Declare single dimensional array and accept 5 integer values from the user. Then sort the input in ascending order and display output.**

| **using System;  class Q2 {  static void Main()  {  //sorting element  int size;  Console.Write("Enter the size of the array: ");  try { size = Convert.ToInt32(Console.ReadLine()); }  catch  {  Console.WriteLine("Invalid input! Defaulting size to 10.");  size = 10;  }  if (size <= 0)  {  Console.WriteLine("Size must be a positive integer. Defaulting to 10.");  size = 10;  }  int[] arr = new int[size];  Console.WriteLine("Enter elements in the array:");  for (int i = 0; i < arr.Length; i++)  {  try  {  Console.Write($"Element {i + 1}: ");  arr[i] = Convert.ToInt32(Console.ReadLine());  }  catch (FormatException)  {  Console.WriteLine("Invalid input! Please enter a valid integer.");  i--; // Decrement i to repeat this iteration  }  }  //soring arry without built-in sort method  for (int i = 0; i < arr.Length - 1; i++)  {  for (int j = i + 1; j < arr.Length; j++)  {  if (arr[i] > arr[j])  {  // Swap arr[i] and arr[j]  int temp = arr[i];  arr[i] = arr[j];  arr[j] = temp;  }  }  }  Console.WriteLine("Sorted array:");    for (int i = 0; i < arr.Length; i++)  {  Console.Write(arr[i] + " ");  }   }  }** |
| --- |

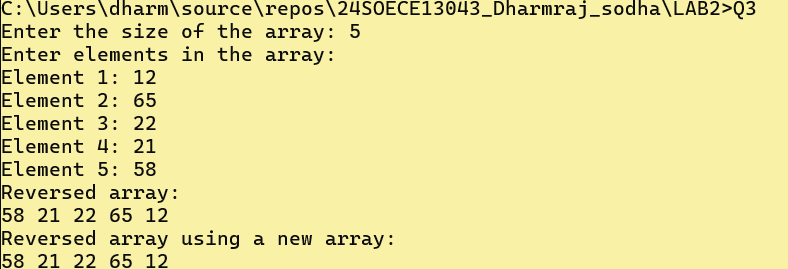
**Output**

****

**3. Write a C# Sharp program to read n values in an array and display them in reverse order.**

| **using System; using System.Runtime.InteropServices;  class Q3 {  static void Main()  {  //revers arrary order  int size;  Console.Write("Enter the size of the array: ");   try  {  size = Convert.ToInt32(Console.ReadLine());  if (size <= 0)  {  Console.WriteLine("Size must be a positive integer. Defaulting to 10.");  size = 10;  }  }  catch (FormatException)  {  Console.WriteLine("Invalid input! Defaulting size to 10.");  size = 10;  }  int[] arr = new int[size];  Console.WriteLine("Enter elements in the array:");  for (int i = 0; i < arr.Length; i++)  {  try  {  Console.Write($"Element {i + 1}: ");  arr[i] = Convert.ToInt32(Console.ReadLine());  }  catch (FormatException)  {  Console.WriteLine("Invalid input! Please enter a valid integer.");  i--; // Decrement i to repeat this iteration  }  }  Console.WriteLine("Reversed array:");  int[] reversedArr = new int[size];  for (int i = arr.Length - 1; i >= 0; i--)  {  reversedArr[arr.Length - 1 - i] = arr[i];  Console.Write(arr[i] + " ");  }    } }** |
| --- |

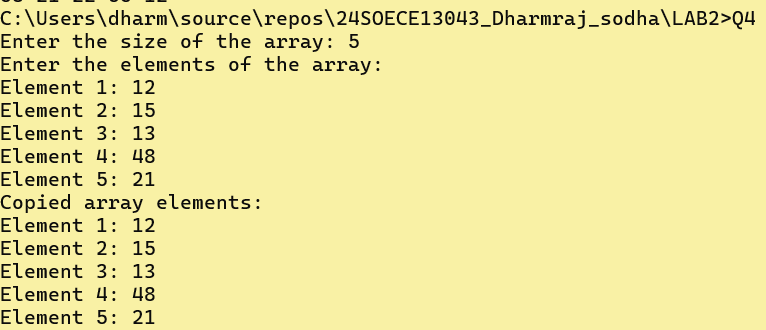
**Output**

****

**4. Write a C# Sharp program to copy the elements of one array into another array.**

| **using System; using System.Reflection;  class Q4 {  static void Main()  {  // copy one arry to another  int size;  Console.Write("Enter the size of the array: ");  size = Convert.ToInt32(Console.ReadLine());  int[] arr1 = new int[size];  int[] arr2 = new int[size];  Console.WriteLine("Enter the elements of the array:");  for (int i = 0; i < size; i++)  {  Console.Write("Element {0}: ", i + 1);  arr1[i] = Convert.ToInt32(Console.ReadLine());  }  // Copying arr1 to arr2  for (int i = 0; i < size; i++)  {  arr2[i] = arr1[i];  }  // Displaying the copied array  Console.WriteLine("Copied array elements:");  for (int i = 0; i < size; i++)  {  Console.WriteLine("Element {0}: {1}", i + 1, arr2[i]);  }  } }** |
| --- |

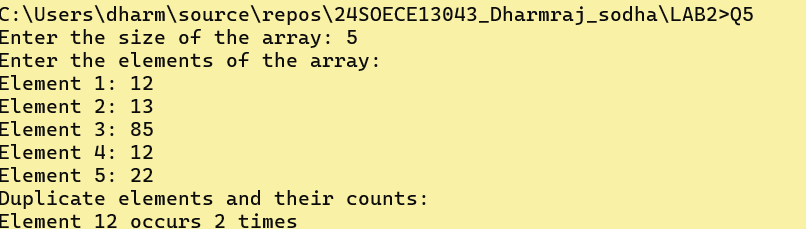
**Output**

****

**5. Write a C# Sharp program to count duplicate elements in an array.**

| **using System; using System.ComponentModel;  class Q5 {  static void Main()  {  int size;  Console.Write("Enter the size of the array: ");  size = Convert.ToInt32(Console.ReadLine());  int[] arr = new int[size];  Console.WriteLine("Enter the elements of the array:");  for (int i = 0; i < size; i++)  {  Console.Write("Element {0}: ", i + 1);  arr[i] = Convert.ToInt32(Console.ReadLine());  }  int[] count = new int[size];  for (int i = 0; i < size; i++)  {  count[i] = 1;   for (int j = i + 1; j < size; j++)  {  if (arr[i] == arr[j])  {  count[i]++;  arr[j] = -1;  }  }  }  Console.WriteLine("Duplicate elements and their counts:");    for (int i = 0; i < size; i++)  {  if (arr[i] != -1 && count[i] > 1)   {  Console.WriteLine("Element {0} occurs {1} times", arr[i], count[i]);  }  }    } }** |
| --- |

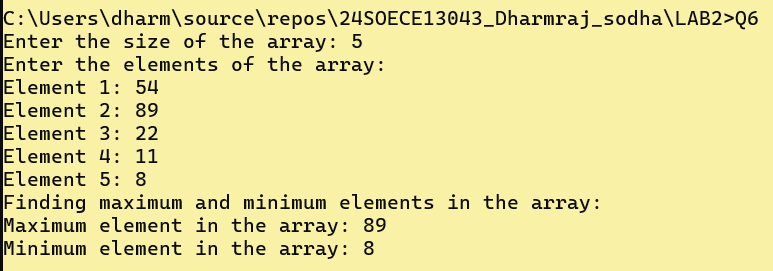
**Output**

****

**6. Write a C# Sharp program to find the maximum and minimum elements in an array.**

| **using System; using System.ComponentModel;  class Q6 {  static void Main()  {  int size;  Console.Write("Enter the size of the array: ");  size = Convert.ToInt32(Console.ReadLine());  int[] arr = new int[size];  Console.WriteLine("Enter the elements of the array:");  for (int i = 0; i < size; i++)  {  Console.Write("Element {0}: ", i + 1);  arr[i] = Convert.ToInt32(Console.ReadLine());  }  // Initialize max and min with the first element  int max = arr[0];  int min = arr[0];  // Find max and min   Console.WriteLine("Finding maximum and minimum elements in the array:");  for (int i = 1; i < size; i++)  {  if (arr[i] > max)  {  max = arr[i];  }  if (arr[i] < min)  {  min = arr[i];  }  }  // Display the results  Console.WriteLine("Maximum element in the array: " + max);  Console.WriteLine("Minimum element in the array: " + min);   } }** |
| --- |

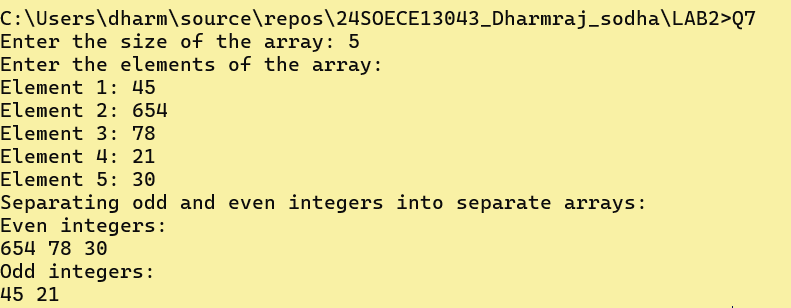
**Output**

****

**7. Write a program in C# Sharp to separate odd and even integers into separate arrays.**

| **using System; using System.ComponentModel;  class Q7 {  static void Main()  {  //Write a program in C# Sharp to separate odd and even integers into separate arrays  int size;  Console.Write("Enter the size of the array: ");  size = Convert.ToInt32(Console.ReadLine());  int[] arr = new int[size];  int evenCount = 0, oddCount = 0;  Console.WriteLine("Enter the elements of the array:");  for (int i = 0; i < size; i++)  {  Console.Write("Element {0}: ", i + 1);  arr[i] = Convert.ToInt32(Console.ReadLine());  if (arr[i] % 2 == 0)  {  evenCount++;  }  else  {  oddCount++;  }  }  // Create separate arrays for even and odd integers  int[] evenArr = new int[evenCount];  int[] oddArr = new int[oddCount];  evenCount = 0; // Reset count for filling even array  oddCount = 0; // Reset count for filling odd array  Console.WriteLine("Separating odd and even integers into separate arrays:");  for (int i = 0; i < size; i++)  {  if (arr[i] % 2 == 0)  {  evenArr[evenCount++] = arr[i]; // Fill even array  }  else  {  oddArr[oddCount++] = arr[i]; // Fill odd array  }  }  // Display the even array  Console.WriteLine("Even integers:");  for (int i = 0; i < evenArr.Length; i++)  {  Console.Write(evenArr[i] + " ");  }  Console.WriteLine();  // Display the odd array  Console.WriteLine("Odd integers:");  for (int i = 0; i < oddArr.Length; i++)  {  Console.Write(oddArr[i] + " ");  }    } }** |
| --- |

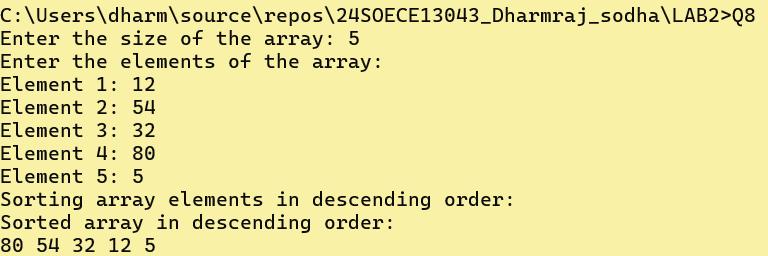
**Output**

****

**8. Write a C# Sharp program to sort array elements in descending order.**

| **using System; using System.ComponentModel;  class Q8 {  static void Main()  {  int size;  Console.Write("Enter the size of the array: ");  size = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter the elements of the array:");  int[] arr = new int[size];  for (int i = 0; i < size; i++)  {  Console.Write("Element {0}: ", i + 1);  arr[i] = Convert.ToInt32(Console.ReadLine());  }  // Sort the array in descending order  Console.WriteLine("Sorting array elements in descending order:");  for (int i = 0; i < size - 1; i++)  {  for (int j = i + 1; j < size; j++)  {  if (arr[i] < arr[j]) // Change to '>' for ascending order  {  // Swap elements  int temp = arr[i];  arr[i] = arr[j];  arr[j] = temp;  }  }  }  // Display the sorted array  Console.WriteLine("Sorted array in descending order:");  for (int i = 0; i < size; i++)  {  Console.Write(arr[i] + " ");  }   } }** |
| --- |

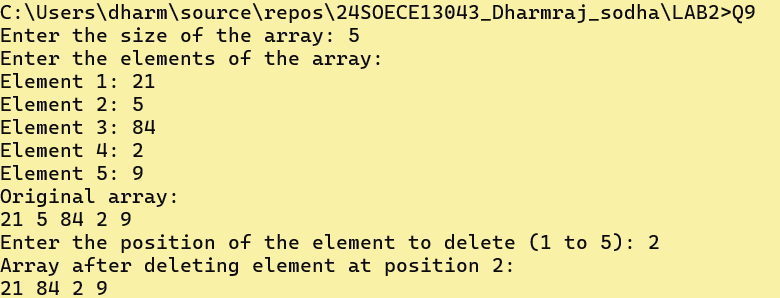
**Output**

****

**9. Write a C# Sharp program to delete an element at the desired position from an array.**

| **using System; using System.ComponentModel;  class Q9 {  static void Main()  {  int size;  Console.Write("Enter the size of the array: ");  size = Convert.ToInt32(Console.ReadLine());  int[] arr = new int[size];  Console.WriteLine("Enter the elements of the array:");  for (int i = 0; i < size; i++)  {  Console.Write("Element {0}: ", i + 1);  arr[i] = Convert.ToInt32(Console.ReadLine());  }  // Display the original array  Console.WriteLine("Original array:");  for (int i = 0; i < size; i++)  {  Console.Write(arr[i] + " ");  }  Console.WriteLine();  // Ask for the position to delete  Console.Write("Enter the position of the element to delete (1 to {0}): ", size);  int position = Convert.ToInt32(Console.ReadLine());  // Validate position  if (position < 1 || position > size)  {  Console.WriteLine("Invalid position! Please enter a position between 1 and {0}.", size);  return;  }  // Adjust position for zero-based index  position--; // Convert to zero-based index  // Create a new array with one less size  int[] newArr = new int[size - 1];  // Copy elements to the new array, skipping the element at the specified position  for (int i = 0, j = 0; i < size; i++)  {  if (i != position) // Skip the element at the specified position  {  newArr[j++] = arr[i];  }  }  // Display the new array  Console.WriteLine("Array after deleting element at position {0}:", position + 1);  for (int i = 0; i < newArr.Length; i++)  {  Console.Write(newArr[i] + " ");  }    } }** |
| --- |

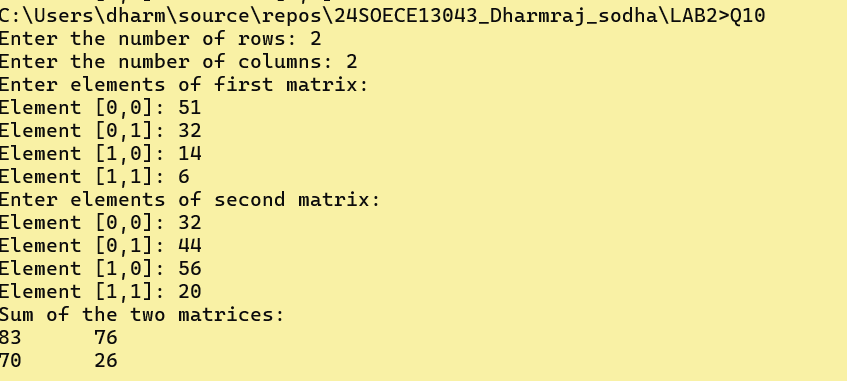
**Output**

****

**10. Write a C# Sharp program for adding two matrices of the same size.**

| **using System; using System.ComponentModel;  class Q10 {  static void Main()  {  //Write a C# Sharp program for adding two matrices of the same size.  int rows, cols;  Console.Write("Enter the number of rows: ");  rows = Convert.ToInt32(Console.ReadLine());  Console.Write("Enter the number of columns: ");  cols = Convert.ToInt32(Console.ReadLine());  int[,] matrix1 = new int[rows, cols];  int[,] matrix2 = new int[rows, cols];  int[,] sumMatrix = new int[rows, cols];  Console.WriteLine("Enter elements of first matrix:");  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  Console.Write($"Element [{i},{j}]: ");  matrix1[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  Console.WriteLine("Enter elements of second matrix:");  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  Console.Write($"Element [{i},{j}]: ");  matrix2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  // Adding the two matrices  Console.WriteLine("Sum of the two matrices:");  for (int i = 0; i < rows; i++)  {  for (int j = 0; j < cols; j++)  {  sumMatrix[i, j] = matrix1[i, j] + matrix2[i, j];  Console.Write(sumMatrix[i, j] + "\t");  }  Console.WriteLine();  }  } }** |
| --- |

**Output**

****