

**1. Write a C# program to find the sum of all elements in an integer array using a loop.**

```
using System;

class Program
{
    static void Main(string[] args)
    {
        int[] nums = { 10, 20, 30, 40, 50 };

        int sum = 0;

        foreach (int num in nums)
        {
            sum += num;
        }

        Console.WriteLine("Sum of array elements: " + sum);
    }
}
```

**2. Create a C# program that calculates the average of values in a floating-point array using a loop.**

```
using System;

namespace Array
{
    class Program
    {
        static void Main(string[] args)
        {
            float[] values = { 2.5f, 3.7f, 1.2f, 4.0f, 5.8f };

            float sum = 0;
```

```

        for (int i = 0; i < values.Length; i++)
        {
            sum += values[i];
        }

        float average = sum / values.Length;

        Console.WriteLine($"The average of the values is: {average}");
    }
}

```

**3. Develop a C# program that finds the largest element in an integer array using a loop and if-else statements.**

```

using System;

namespace Largest
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] nums = { 10, 3, 45, 27, 8, 50 };

            int large = nums[0];

            for (int i = 1; i < nums.Length; i++)
            {
                if (nums[i] > large)
                {
                    large = nums[i];
                }
            }
        }
    }
}

```

```

    }

    Console.WriteLine($"The largest element in the array is: {large}");

}

}

}

```

**4. Write a C# program that counts the number of even and odd elements in an integer array using a loop and if-else statements.**

```
using System;
```

```
namespace EvenOdd
```

```

{

    class Program

    {

        static void Main(string[] args)

        {

            int[] numbers = { 10, 3, 45, 27, 8, 50};

            int evenCount = 0;

            int oddCount = 0;

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

            {

                if (numbers[i] % 2 == 0)

                {

                    evenCount++;

                }

                else

                {

```

```

        oddCount++;
    }
}

Console.WriteLine($"Number of even elements: {evenCount}");

Console.WriteLine($"Number of odd elements: {oddCount}");

}

}

}

```

**5. Implement a C# program that reverses the elements of an integer array using a loop.**

```

using System;

namespace ArrayReverse
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] numbers = { 10, 3, 45, 27, 8, 50 };

            int startIndex = 0;

            int endIndex = numbers.Length - 1;

            while (startIndex < endIndex)
            {
                int temp = numbers[startIndex];

                numbers[startIndex] = numbers[endIndex];

                numbers[endIndex] = temp;

                startIndex++;
            }
        }
    }
}

```

```

        endIndex--;
    }

    Console.WriteLine("Reversed array:");

    foreach (int num in numbers)
    {
        Console.Write(num + " ");
    }
}
}

```

**6.Create a C# program that multiplies each element in an integer array by a specified factor using a loop.**

```

using System;

namespace Array
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] numbers = { 10, 3, 45, 27, 8, 50 };

            int factor = 2;

            for (int i = 0; i < numbers.Length; i++)
            {
                numbers[i] *= factor;
            }

            Console.WriteLine("Modified array:");

```

```

        foreach (int num in numbers)
        {
            Console.Write(num + " ");
        }
    }
}

```

**7. Write a C# program that searches for a specific value in an integer array using a loop and returns its index if found.**

```

using System;

namespace ArraySearch
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] numbers = { 10, 3, 45, 27, 8, 50 };

            int targetValue = 27;

            int index = -1;

            for (int i = 0; i < numbers.Length; i++)
            {
                if (numbers[i] == targetValue)
                {
                    index = i;

                    break;
                }
            }
        }
    }
}

```

```

    }

    if (index != -1)
    {
        Console.WriteLine($"Value found at index {index}");
    }

    else
    {
        Console.WriteLine($"Value not found in the array");
    }
}
}
}

```

**8. Develop a C# program that finds the second smallest element in an integer array using loops and sorting techniques.**

```
using System;
```

```
namespace Second
```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            int[] numbers = { 10, 3, 45, 27, 8, 50 };

            Array.Sort(numbers);

            int secondSmallest = -1;

            for (int i = 1; i < numbers.Length; i++)

```

```

    {
        if (numbers[i] != numbers[i - 1])
        {
            secondSmallest = numbers[i];
            break;
        }
    }

    if (secondSmallest != -1)
    {
        Console.WriteLine($"The second smallest element is: {secondSmallest}");
    }
    else
    {
        Console.WriteLine("No second smallest element found in the array.");
    }
}
}
}

```

**9. Create a C# program that removes all duplicates from an integer array using loops and additional data structures.**

```

using System;

using System.Collections.Generic;

namespace RemoveDupli
{
    class Program

```



```

{
    static void Main(string[] args)
    {
        int[] numbers = { 10, 3, 45, 27, 8, 50, 3, 10, 27 };

        HashSet<int> uniqueNumbers = new HashSet<int>();

        for (int i = 0; i < numbers.Length; i++)
        {
            uniqueNumbers.Add(numbers[i]);
        }

        int[] resultArray = new int[uniqueNumbers.Count];
        uniqueNumbers.CopyTo(resultArray);

        Console.WriteLine("Array without duplicates:");

        foreach (int num in resultArray)
        {
            Console.Write(num + " ");
        }
    }
}

```

**10. Write a C# program that finds the common elements between two integer arrays using loops.**

```
using System;
```

```
namespace Common
```

```
{
```

```
    class Program
```

```
{  
    static void Main(string[] args)  
    {  
        int[] array1 = { 10, 3, 45, 27, 8, 50 };  
        int[] array2 = { 27, 8, 12, 3, 35 };  
        Console.WriteLine("Common elements:");  
        foreach (int num1 in array1)  
        {  
            foreach (int num2 in array2)  
            {  
                if (num1 == num2)  
                {  
                    Console.Write(num1 + " ");  
                    break;  
                }  
            }  
        }  
    }  
}
```