```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace project_1
{
    internal class Program
    {
        class Statistics
        {
            private int[] A;
            private int n;

            public Statistics(int[] items)
            {
                this.A = items;
                this.n = items.Length;
            }

            public double Median()
            {
                Array.Sort(A);
                double median;
                if (n % 2 == 0)
                {
                    median = (A[n / 2] + A[(n / 2) - 1]) / 2.0;
                }
                else
                {
                    median = A[n / 2];
                }
                return median;
            }

            public int Mode()
            {
                int[] counts = new int[A.Max() + 1];
                int mode = 0;
                int maxCount = 0;

                foreach (int item in A)
                {
                    counts[item]++;
                }

                for (int i = 0; i < counts.Length; i++)
                {
                    if (counts[i] > maxCount)
                    {
```

```
            mode = i;
            maxCount = counts[i];
        }
    }

    return mode;
}
public int Range()
{
    int max = A.Max();
    int min = A.Min();
    int range = max - min;
    return range;
}
private double MedianHelper(int[] arr)
{
    Array.Sort(arr);
    double median;
    int len = arr.Length;
    if (len % 2 == 0)
    {
        median = (arr[len / 2] + arr[(len / 2) - 1]) / 2.0;
    }
    else
    {
        median = arr[len / 2];
    }
    return median;
}

public double FirstQuartile()
{
    int mid = n / 2;
    double q1;
    if (n % 2 == 0)
    {
        q1 = MedianHelper(A.Take(mid).ToArray());
    }
    else
    {
        q1 = MedianHelper(A.Take(mid + 1).ToArray());
    }
    return q1;
}

public double ThirdQuartile()
{
    int mid = n / 2;
    double q3;
    if (n % 2 == 0)
    {
```

```
                q3 = MedianHelper(A.Skip(mid).Take(mid).ToArray());
            }
            else
            {
                q3 = MedianHelper(A.Skip(mid + 1).Take(mid).ToArray());
            }
            return q3;
        }

        public int P90()
        {
            double percentile = n * 0.9;
            int p90 = A[(int)percentile - 1];
            return p90;
        }

        public double InterquartileRange()
        {
            double q1 = FirstQuartile();
            double q3 = ThirdQuartile();
            double iqr = q3 - q1;
            return iqr;
        }

        public (int, int) OutlierBoundaries()
        {
            double q1 = FirstQuartile();
            double q3 = ThirdQuartile();
            double iqr = InterquartileRange();
            int lowerBound = (int)(q1 - (1.5 * iqr));
            int upperBound = (int)(q3 + (1.5 * iqr));
            return (lowerBound, upperBound);
        }

        public bool IsOutlier(int value)
        {
            (int lowerBound, int upperBound) = OutlierBoundaries();
            bool isOutlier = value < lowerBound || value > upperBound;
            return isOutlier;
        }


        public void print()
        {
            Console.WriteLine("Median: " + Median());
            Console.WriteLine("Mode: " + Mode());
            Console.WriteLine("Range: " + Range());
            Console.WriteLine("First Quartile: " + FirstQuartile());
            Console.WriteLine("Third Quartile: " + ThirdQuartile());
            Console.WriteLine("P90: " + P90());
```

```
                Console.WriteLine("Interquartile Range: " +
InterquartileRange());

                (int lowerBound, int upperBound) = OutlierBoundaries();
                Console.WriteLine("Outlier Boundaries: " + lowerBound + " to
" + upperBound);
            }
        }
        static void Main(string[] args)
        {

                Console.Write("Enter the number of items: ");
                int n = int.Parse(Console.ReadLine());

                int[] items = new int[n];
                Console.WriteLine("Enter the item values:");
                for (int i = 0; i < n; i++)
                {
                Console.WriteLine("enter value naumber {0}",i+1);
                items[i] = int.Parse(Console.ReadLine());
                }

                Statistics stats = new Statistics(items);

                stats.print();

                Console.Write("Enter a value to check for outlier: ");
                int checkValue = int.Parse(Console.ReadLine());
                bool isOutlier = stats.IsOutlier(checkValue);
                Console.WriteLine("Is " + checkValue + " an outlier? " +
isOutlier);
            Console.ReadKey();

        }
    }
}
```