using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Runtime.Remoting.Messaging;

using System.Text;

using System.Threading.Tasks;

namespace proptask

{

internal class Program

{

static void Main(string[] args)

{

int i, n;

Console.WriteLine(" Input the number of elements to store in the array: ");

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

float[] arr = new float[(int)n];

Console.WriteLine("\nInput the values of the {0} elements of the array: ", n);

for (i = 0; i < n; i++)

{

Console.Write(" element {0}: ", i + 1);

arr[(int)i] = float.Parse(Console.ReadLine());

}

Console.WriteLine("-----------------------------------------------------------------------------");

// (1) The median

if (n % 2 == 0)

{

float mideleelement1 = (arr[(n / 2) - 1]);

float mideleelement2 = (arr[n / 2]);

Console.WriteLine(" The Median = {0} , {1}", mideleelement1, mideleelement2);

}

else

{

float median = arr[n / 2];

Console.WriteLine(" The Median = {0}", median);

}

// (2) The mode

float mode = arr[0];

int maxcount = 1;

int currentcount = 1;

for (int x = 1; x < arr.Length; x++)

{

if (arr[x] == arr[x - 1])

{

currentcount++;

}

else

{

if (currentcount >= maxcount)

{

maxcount = currentcount;

mode = arr[x - 1];

}

currentcount = 1;

}

}

if (currentcount > maxcount)

{

mode = arr[arr.Length - 1];

}

Console.WriteLine("\n The mode of the array = " + mode);

// (3) The range

double maxvalue = arr.Max();

double minvalue = arr.Min();

double range = maxvalue - minvalue;

Console.WriteLine("\n The Range = {0}", range);

// (4) The first Quartile

double Q1 = ((float)n + 1)/4;

Console.WriteLine("\n The first Quartile = {0}", Q1);

// (5) The third Quartile

double Q3 = ( (float) n + 1) \*((float)3/4);

Console.WriteLine(" The third Quartile = {0}", Q3);

// (6) The P90

double p90Index = ((arr.Length \* 0.9) - 1);

double p90 = arr[(int)p90Index];

Console.WriteLine("\n P90 = {0}", p90);

// (7) The interquartile

double IQR = Q3 - Q1;

Console.WriteLine("\n The Interquartile = {0}", IQR);

// (8) The boundaries of the outlier

double lowerOutlierBoundary = (Q1 - (1.5 \* IQR));

Console.WriteLine("\n The Lower outlier boundary = {0}", lowerOutlierBoundary);

double upperOutlierBoundary = (Q3 + (1.5 \* IQR));

Console.WriteLine(" The Upper outlier boundary = {0}\n", upperOutlierBoundary);

// (9) Determine whether an input value is an outlier or not

for (int x = 0; x < n; x++)

{

if( x == Q1 || x == Q3)

Console.WriteLine(" element {0} is an outlier " , x+1 );

else

continue;

}

}

}

}