```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Globalization;
 4  using System.Linq;
 5  using System.Security.Cryptography.X509Certificates;
 6  using System.Text;
 7  using System.Threading.Tasks;
 8
 9  namespace probability_project
10  {
11      internal class Program
12      {
13
14          static void Main(string[] args)
15          {
16              Console.WriteLine("Enter the number of elements");
17
18
19              // you enter the group of data, where n is the number of
                  elements
20              int n = int.Parse(Console.ReadLine());
21              double[] array = new double[n];
22
23
24              Console.WriteLine("-_-_-_-_-_-_-_-_-_-_-_-_-");
25
26
27              // (var) is the index of the minimum value, Because (i)
                  starts from 0
28              int var = 0;
29              for(int i = 0; i < n; i++)
30              {
31                  Console.WriteLine("Enter value number"+(i+1));
32                  array[i]=int.Parse(Console.ReadLine());
33              }
34
35
36
37              // (s) is the index of the maximum value
38              int s = array.Length - 1;
39
40
41
42              // we should arrange this elements
43              Array.Sort(array);
44
45
46
47              Console.WriteLine("-_-_-_-_-_-_-_-_-_-_-_-_-");
48              Console.WriteLine("Compute the Range");
49
50
51              //print minimum and maximum value
```

```
52              Console.WriteLine("Minimum value is "+ array[var]);
53              Console.WriteLine("Maximum value is "+ array[s]);
54
55
56              //find the Range form this formula : ( Range = Max - Min )
57              double Range = array[s] - array[var];
58
59              //Print the Range
60              Console.WriteLine("The Range = "+ Range);
61
62
63              Console.WriteLine("-_-_-_-_-_-_-_-_-_-_-_-_-_-");
64
65
66
67              //$ compute the median $
68              Console.WriteLine("Compute the median");
69              int k = array.Length;
70              double median = 0;
71              if(k%2==0)
72              {
73                  // if the number of data is even ( this formula : sum
                      of two numbers in the middle, then sum  divide to 2
74                   median = (array[k / 2] / 2 + array[(k / 2) - 1] / 2);
75                  Console.WriteLine("The number of this elements are
                      even");
76
77
78                  //print the median
79                  Console.WriteLine("The median = "+ median);
80              }
81              else
82              {
83                  //if the number of data is odd (this formula : the
                      number in the middle
84                  median = array[k / 2];
85                  Console.WriteLine("The number of this elements are
                      odd");
86
87
88                  //print the median
89                  Console.WriteLine("The median/Q2 = "+ median);
90              }
91
92              Console.WriteLine("-_-_-_-_-_-_-_-_-_-_-_-_-_-");
93
94
95              //compute Q1(first quartile
96              double quartile1 = 0;
97              int g = array.Length / 2;
98              for (int i = 0; i < n; i++)
99              {
100
```

```
101                 if(g%2==0)
102                 {
103                     quartile1 = array[( g/2)-1]/2 +array[( g/2)]/2;
104                 }
105                 else
106                 {
107                     quartile1 = array[g / 2];
108                 }
109             }
110             Console.WriteLine("Q1 = "+quartile1);
111
112
113
114
115
116             //compute Q3 (third quartile)
117             double quartile3 = 0;
118             for( int i = 0; i < n && n%2==0; i++)
119             {
120                 if (g % 2 == 0)
121                 {
122                     quartile3 = array[(3 * g/2)-1]/2+array[(3 *
                      g/2)]/2;
123                 }
124                 else
125                 {
126                     quartile3 = array[(3 * g / 2)+1];
127                 }
128             }
129             for(int i = 0; i < n && n%2==1; i++)
130             {
131                 if (g % 2 == 0)
132                 {
133                     quartile3 = array[(3 * g / 2)] / 2 + array[(3 * g /
                      2) + 1] / 2;
134                 }
135                 else
136                 {
137                     quartile3 = array[(3 * g / 2) + 1];
138                 }
139             }
140             Console.WriteLine("Q3 = "+quartile3);
141
142
143
144             //compute IQR (IQR = Q3 -Q1)
145             double IQR = 0;
146             IQR = quartile3 - quartile1;
147             Console.WriteLine("IQR = "+IQR);
148
149
150             //compute the outlier (outlier : [Q1 -1.5 * IQR , Q3 +1.5 *
                  IQR] )
```

```
151             Console.WriteLine("Outlier : "+ (quartile1 -1.5 *IQR ,
                  quartile3 +1.5 * IQR));
152
153
154             //check the value is outlier or not
155             Console.WriteLine("Enter the value to check if it finds in
                  interval or not");
156             double value = double.Parse(Console.ReadLine());
157             if(value > quartile3 +1.5 * IQR|| value < quartile1 - 1.5 *
                   IQR)
158             {
159                 Console.WriteLine("This value doesn't belong to the
                      interval, then it's outlier");
160             }
161
162
163             else
164             {
165                 Console.WriteLine("This value belongs to the interval,
                      then it isn't outlier");
166             }
167
168             Console.WriteLine("-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-_-");
169
170
171             //compute percentile
172             Console.WriteLine("Enter the value of p");
173             int p = int.Parse(Console.ReadLine());
174             double u = (double)p / 100;
175             Console.WriteLine("u = "+u);
176             double position = u*n ;
177             Console.WriteLine("position = "+position);
178
179             double percentile = 0;
180
181             if (position % 1==0)
182             {
183                 Console.WriteLine("the number is integer");
184                 Console.WriteLine("index = "+(position,position+1));
185                 percentile = array[((int)position)-1]/2 +array[((int)
                      position)]/2;
186             }
187             else
188             {
189
190                 Console.WriteLine("the number is fraction");
191                 Console.WriteLine("index = "+((int)position+1));
192                 percentile = array[((int)position) ];
193
194             }
195
196             Console.WriteLine("percentile = "+percentile);
197
```

```
198
199
200             //compute the mode
201            // Declare a variable to store the mode
202             double mode = 0;
203
204             // Declare a variable to store the maximum frequency
205             int maxFrequency = 0;
206
207             // Loop through the distinct items in the array
208             foreach (double item in array.Distinct())
209             {
210                 // Count the frequency of the current item in the array
211                 int frequency = array.Count(x => x == item);
212
213                 // If the current frequency is higher than the maximum
                      frequency, update the mode and the maximum frequency
214                 if (frequency > maxFrequency)
215                 {
216                     mode = item;
217                     maxFrequency = frequency;
218                 }
219             }
220
221             // Check if there is any duplicate number entered
222             if (maxFrequency > 1)
223             {
224                 // Display the mode
225                 Console.WriteLine("The mode of the items is {0}",
                      mode);
226             }
227             else
228             {
229                 // Display zero
230                 Console.WriteLine("Mode = "+mode);
231             }
232
233
234
235
236
237
238             Console.ReadKey();
239         }
240
241
242
243     }
244 }
245
```