using System;

using System.Collections.Generic;

namespace project\_1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("choose program from 1 to 9");

Console.WriteLine("1-The median of the values ");

Console.WriteLine("2-The mode of the values ");

Console.WriteLine("3-The range of the values ");

Console.WriteLine("4-The first Quartile of the values ");

Console.WriteLine("5-The third Quartile of the values");

Console.WriteLine("6-The P90 of the values");

Console.WriteLine("7-The interquartile of the values");

Console.WriteLine("8-The boundaries of the outlier region");

Console.WriteLine("9-Determine whether an input value is an outlier or not.");

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

switch (program)

{

case 1:

//nuber 1 The median of the values Ai

Console.WriteLine("enter number of values");

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

float[] n = new float[a];

if (a > 0)

{

for (int i = 0; i < a; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

if (a % 2 != 0)

{

Array.Sort(n);

Console.WriteLine("number is " + n[(a / 2) + (1 / 2)]);

}

else

{

float m = n[(a / 2)] + n[a / 2 - 1];

m = m / 2;

Console.WriteLine("number is " + m);

}

}

else

{

Console.WriteLine("Error start with 1");

}

break;

case 2:

Console.WriteLine("enter number of values");

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

int[] q = new int[f];

if (f > 0)

{

for (int i = 0; i < f; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

var dict = new Dictionary<int, int>();

foreach (var value in q)

{

if (dict.ContainsKey(value))

dict[value]++;

else

dict[value] = 1;

}

foreach (var pair in dict)

{

Console.WriteLine("{0} = {1} time(s)", pair.Key, pair.Value);

}

}

else

Console.WriteLine("Error");

break;

case 3:

Console.WriteLine("enter number of values");

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

if (w > 0)

{

int[] e = new int[w];

for (int i = 0; i < w; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

Array.Sort(e);

int p = e[w - 1] - e[0];

Console.WriteLine("Numper is " + p);

}

else

Console.WriteLine("Error");

break;

case 4:

Console.WriteLine("enter number of values");

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

int[] k = new int[d];

if (d > 0)

{

for (int i = 0; i < d; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

if (d % 2 != 0)

{

Array.Sort(k);

d = d + 1;

d = d / 4;

Console.WriteLine("number is " + k[d - 1]);

}

else

{

d = d + 1;

int m = (d / 4);

float u = k[m - 1] + k[m];

Console.WriteLine("number is " + u / 2);

}

}

break;

case 5:

Console.WriteLine("enter number of values");

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

float[] s = new float[z];

if (z > 0)

{

for (int i = 0; i < z; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

if (z % 2 != 0)

{

Array.Sort(s);

z = z + 1;

z = (z \* 3) / 4;

Console.WriteLine("number is " + s[z - 1]);

}

else

{

z = z + 1;

int m = (z \* 3);

m = (m / 4);

float u = s[m - 1] + s[m];

Console.WriteLine("number is " + u / 2);

}

}

break;

case 6:

Console.WriteLine("enter number of values");

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

int[] gg = new int[cc];

if (cc > 0)

{

for (int i = 0; i < cc; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

float p90 = (90 \* (cc + 1)) / 100;

Console.WriteLine("p90 = " + p90);

}

else

Console.WriteLine("error");

break;

case 7:

Console.WriteLine("enter number of values");

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

int[] g = new int[c];

if (c > 0)

{

for (int i = 0; i < c; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

if (c % 2 != 0)

{

Array.Sort(g);

int r = c + 1;

r = r / 4;

c = c + 1;

c = (c \* 3) / 4;

Console.WriteLine("number is " + (g[c - 1] - g[r - 1])); //iqr

}

else

{

Array.Sort(g);

int zz = c + 1;

zz = zz / 4;

float q1 = (g[zz] + g[zz - 1]);

q1 = q1 / 2; //q1

int qq = c + 1;

qq = (qq \* 3) / 4;

float q3 = (g[qq] + g[qq - 1]);

q3 = (q3 / 2); //q3

Console.WriteLine("number is " + (q3 - q1)); //iqr

}

}

break;

case 8:

Console.WriteLine("enter number of values");

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

g = new int[c];

if (c > 0)

{

for (int i = 0; i < c; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

if (c % 2 != 0)

{

Array.Sort(g);

int r = c + 1;

r = r / 4; //q1

c = c + 1;

c = (c \* 3) / 4; //q3

float IQR= (g[c - 1] - g[r - 1]); //iqr

Console.WriteLine("lower outlier boundary =" + (r - (3 / 2 \* IQR)));

Console.WriteLine("UPPER outlier boundary =" + (c + (3 / 2 \* IQR)));

}

else

{

Array.Sort(g);

int zz = c + 1;

zz = zz / 4;

float q1 = (g[zz] + g[zz - 1]);

q1 = q1 / 2; //q1

int qq = c + 1;

qq = (qq \* 3) / 4;

float q3 = (g[qq] + g[qq - 1]);

q3 = (q3 / 2); //q3

float IQR = (q3 - q1); //iqr

Console.WriteLine("lower outlier boundary =" + (q1 - (3 / 2 \* IQR)));

Console.WriteLine("UPPER outlier boundary =" + (q3 + (3 / 2 \* IQR)));

}

}

break;

case 9:

Console.WriteLine("enter number of values");

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

g = new int[c];

if (c > 0)

{

for (int i = 0; i < c; i++)

{

Console.WriteLine("Enter number " + (i + 1));

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

}

if (c % 2 != 0)

{

Array.Sort(g);

int r = c + 1;

r = r / 4; //q1

c = c + 1;

c = (c \* 3) / 4; //q3

float IQR = (g[c - 1] - g[r - 1]); //iqr

float lo = (r - (3 / 2 \* IQR));

float up = (c + (3 / 2 \* IQR));

Console.WriteLine("lower outlier boundary =" + lo);

Console.WriteLine("UPPER outlier boundary =" + up);

bool aaa = true;

for(int ii =0; ii < c; ii++)

{

if (g[ii] < up && g[ii] > lo)

{

aaa = true;

}

else

aaa = false;

}

if(aaa == true)

{

Console.WriteLine("Every thimg is Good");

}

else

{

Console.WriteLine("There is some values out of boundary");

}

}

else

{

Array.Sort(g);

int zz = c + 1;

zz = zz / 4;

float q1 = (g[zz] + g[zz - 1]);

q1 = q1 / 2; //q1

int qq = c + 1;

qq = (qq \* 3) / 4;

float q3 = (g[qq] + g[qq - 1]);

q3 = (q3 / 2); //q3

float IQR = (q3 - q1); //iqr

Console.WriteLine("lower outlier boundary =" + (q1 - (3 / 2 \* IQR)));

Console.WriteLine("UPPER outlier boundary =" + (q3 + (3 / 2 \* IQR)));

}

}

break;

default:

Console.WriteLine("sorry doc samir out of choose");

break;

}

}

Name: Mahmoud Adel Taha Hegazy. (SWE).

Name: Yousef Ahmed Osman.(SWE).

<http://www.probablity.com>

   }

}