Day 0: Mean, Median, and Mode

by AvmnuSng

Objective

In this challenge, we practice calculating the *mean*, *median*, and *mode*. Check out the Tutorial tab for learning materials and an instructional video!

Task

Given an array, X, of N integers, calculate and print the respective *mean*, *median*, and *mode* on separate lines. If your array contains more than one *modal value*, choose the numerically smallest one.

Note: Other than the modal value (which will always be an integer), your answers should be in decimal form, rounded to a scale of 1 decimal place (i.e., 12.3, 7.0 format).

Input Format

The first line contains an integer, N, denoting the number of elements in the array.
The second line contains N space-separated integers describing the array's elements.

Constraints

- $10 \leq N \leq 2500$
- $0 < x_i \leq 10^5$, where x_i is the i^{th} element of the array.

Output Format

Print 3 lines of output in the following order:

1. Print the *mean* on a new line, to a scale of 1 decimal place (i.e., 12.3, 7.0).
2. Print the *median* on a new line, to a scale of 1 decimal place (i.e., 12.3, 7.0).
3. Print the *mode* on a new line; if more than one such value exists, print the numerically smallest one.

Sample Input

```
10
64630 11735 14216 99233 14470 4978 73429 38120 51135 67060
```

Sample Output

```
43900.6
44627.5
4978
```

Explanation

Mean:

We sum all N elements in the array, divide the sum by N, and print our result on a new line.

$$\mu = \frac{x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9}{10} = \frac{439006}{10} = 43900.6$$

Median:

To calculate the median, we need the elements of the array to be sorted in either non-increasing or non-decreasing order. The sorted array $X = \{4978, 11735, 14216, 14470, 38120, 51135, 64630, 67060, 73429, 99233\}$. We then average the two middle elements:

$$median = \frac{x_4 + x_5}{2} = \frac{89255}{2} = 44627.5$$

and print our result on a new line.

Mode:

We can find the number of occurrences of all the elements in the array:

```
 4978 : 1
11735 : 1
14216 : 1
14470 : 1
38120 : 1
51135 : 1
64630 : 1
67060 : 1
73429 : 1
99233 : 1
```

Every number occurs once, making 1 the maximum number of occurrences for any number in X. Because we have multiple values to choose from, we want to select the smallest one, 4978, and print it on a new line.

Solved score: 30.00pts

Easy Submitted 17388 times
Max Score **30**

Need Help?

View Tutorial

View Discussions

View Editorial Solution

View Top Submissions

Rate This Challenge:

Download problem statement

Download sample test cases

Suggest Edits

Current Buffer (saved locally, editable) Java 8

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Solution {
5
6     public static void main(String[] args) {
```

```
        /* Enter your code here. Read input from STDIN.
        Print output to STDOUT. Your class should be named Solution. */

        Scanner sc = new Scanner(System.in);
        int x = sc.nextInt();

        double[] n = new double[x];

        for (int i = 0; i < x; i++) {
            n[i] = sc.nextDouble();
        }

        Arrays.sort(n);
        System.out.println(getMean(n, x));
        System.out.println(getMedian(n, x));
        System.out.println(getMode(n, x));

    }

    public static double getMean(double[] n, int x) {

        double mean = 0;

        for (int i = 0; i < x; i++) {
            mean += n[i];
        }
        return mean / x;
    }

    public static double getMedian(double[] n, int x) {

        // if odd
        if (n.length % 2 == 1) {
            return n[n.length / 2];
        } else {
            return (n[n.length/2] + n[(n.length/2)-1]) / 2;
        }

    }

    // get mode: the highest number of occurrences of each element in the array
    // then print the smallest number if mode is 1;
    public static int getMode(double[] n, int x) {

        int frequency = 0; //count the occurrences of ele
        int mode = 1;
        int modeValue = (int) n[0]; // a value of an element

        for (int i = 0; i < x; i++) {
            for (int j = 0; j < x; j++) {
                if (n[i] == n[j]) { //if same ele is found, make ele mode.
                    frequency++; // =2
                }
            }

            if (frequency > mode) {
                modeValue = (int) n[i];
                mode = frequency;
                frequency = 0; // resets frequency
            } else {
                frequency = 0;
            }

        }
        return modeValue;
    }
}

/*

10
64630 11735 14216 99233 14470 4978 73429 38120 51135 67060

```

```
80
81
82
83
84
85  */
```

Line: 67 Col: 31

Upload Code as File Test against custom input Run Code Submit Code

Testcase 0 ✔

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Score: 1

Input (stdin)

```
10
64630 11735 14216 99233 14470 4978 73429 38120 51135 67060
```

Your Output (stdout)

```
43900.6
44627.5
4978
```

Expected Output

```
43900.6
44627.5
4978
```