







Leaderboard









Dashboard > Tutorials > 10 Days of Statistics > Day 0: Mean, Median, and Mode

# Day 0: Mean, Median, and Mode



Tutorial Problem Submissions Leaderboard Discussions Editorial

# 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

- $10 \le N \le 2500$
- $0 < x_i \le 10^5$  , 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  $m{N}$  elements in the array, divide the sum by  $m{N}$ , 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:

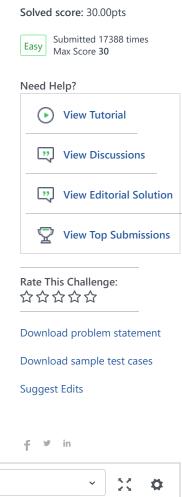
$$median = \frac{x_4 + x_5}{2} = \frac{89255}{2} = 44627.5$$

and print our result on a new line.

#### Mode:

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

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.





```
7 ▼
             /* Enter your code here. Read input from STDIN.
8
             Print output to STDOUT. Your class should be named Solution. */
9
10
            Scanner sc = new Scanner(System.in);
11
             int x = sc.nextInt();
12
            double[] n = new double[x];
13 ▼
14
15 ▼
            for (int i = 0; i < x; i++) {
16 ▼
                n[i] = sc.nextDouble();
17
18
19
            Arrays.sort(n);
20
            System.out.println(getMean(n, x));
21
            System.out.println(getMedian(n, x));
22
            System.out.println(getMode(n, x));
23
24
25
26 ▼
        public static double getMean(double[] n, int x) {
27
28
            double mean = 0;
29
30
            for (int i = 0; i < x; i++) {
31
                mean += n[i];
32
33
            return mean / x;
34
35
        public static double getMedian(double[] n, int x) {
36 ▼
37
             // if odd
38
39
            if (n.length % 2 == 1) {
40
                 return n[n.length / 2];
41 •
            } else {
                return (n[n.length/2] + n[(n.length/2)-1]) / 2;
42
43
44
45
46
47
        // get mode: the highest number of occurrences of each element in the array
48
        // then print the smallest number if mode is 1;
49
        public static int getMode(double[] n, int x) {
50
            int frequency = 0; //count the occurrences of ele
51
52
            int mode = 1;
53 ▼
            int modevalue = (int) n[0]; // a value of an element
54
            for (int i = 0; i < x; i++) {
55 ▼
                 for (int j = 0; j < x; j++) {
56 ▼
57
                     if (n[i] == n[j]) { //if same ele is found, make ele mode.
58
                         frequency++; // =2
59
60
61
                if (frequency > mode) {
62 ▼
                     modevalue = (int) n[i];
63 ▼
64
                     mode = frequency;
65
                     frequency = 0; // resets frequency
66
                } else {
67
                     frequency = 0;
68
69
70
71
            return modeValue;
72
73
    }
74
75
76
77
    64630 11735 14216 99233 14470 4978 73429 38120 51135 67060
78
79
```

Join us on IRC at #hackerrank on freenode for hugs or bugs.

Contest Calendar | Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature