Open-Source Software Practice
Assignment 1

Instructor: Jaemin Jo (`jmjo@skku.edu`)

October 17, 2022

1 Assignment Description

We developed a command-line tool, `stat`. Let's improve it by adding more commands.

2 Add the `med` command (30 pts)

Add a new command, `med`, which computes the median of the given array. If there are an odd number of elements, print out the middle element when the array is sorted. If there are an even number of elements, print out the mean between the elements just before and after the middle point. See the examples below:

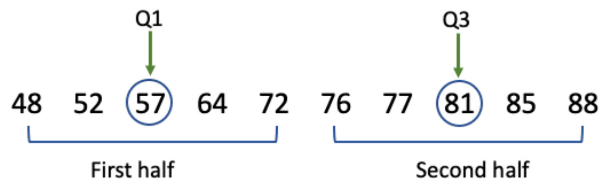
```
> node main.js med 5 1 2 3 3
3
> node main.js med 1 2 4 4
3
```

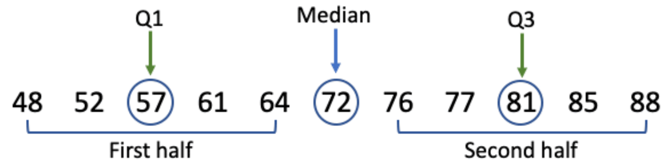
In the first case, we have $[1, 2, 3, 3, 5]$ (sorted), so the middle element is 3. In the second case, we have $[1, 2, 4, 4]$. Since the array has an even number of elements, we compute the mean of the elements before/after the middle point, 2 and 4, so the result is 3. Your answer will be considered correct if the difference between yours and the correct one is less than 1.0.

3 Add the `iqr` command (30 pts)

Add a new command, `iqr`, which computes the interquartile range (IQR) of the given array. IQR is defined as the difference between the 3rd quartile ($Q3$) and the 1st quartile ($Q1$), $IQR = |Q3 - Q1|$

To compute $Q1$ and $Q3$, we need to sort the array in ascending order and split them into two subarrays, S1 and S2. If there are an even number of elements, you can split the sorted array into two equal-length pieces S1 and S2. If there are an odd number of elements, S1 and





S2 are defined as the numbers that are on the left or right side of the median element (the median itself is not inlcuded in either subarray.), respectively. See the examples above.

$Q1$ and $Q3$ are defined as the median of S1 and S2, respectively. Given an array of numbers, compute and print out IQR. Your answer will be considered correct if the difference between yours and the correct one is less than 1.0. It is guaranteed that the length of the input array is equal to or longer than 2.

```
> node main.js iqr 48 52 57 64 72 76 77 81 85 88
24
> node main.js iqr 2 4.5 3
2.5
```

4 Add the `outlier` command (40 pts)

Add a new command, `outlier`, which prints out the outliers in the given array. A number is an outlier if it is less than $Q1 - 1.5 \cdot IQR$ or greater than $Q3 + 1.5 \cdot IQR$. Print out the outliers in the order that they appear in the input. It is guaranteed that the length of the input array is equal to or longer than 2. If there is no outlier, your program should print nothing.

```
> node main.js outlier 1 2 3 4 100
> node main.js outlier 1 1 1 1 1 1 1 1 1 101 100
101
100
> node main.js outlier -17 -5 -2 1 2 3 4
> node main.js outlier -18 -5 -2 1 2 3 4
-18
```

For the input $[-17, -5, -2, 1, 2, 3, 4]$, $Q1$ is -5 and $Q3$ is 3. So, IQR is 8. The lower bound for an outlier is thus -17 ($Q1 - 1.5 \cdot IQR = -5 - 1.5 \cdot 8 = -17$). Therefore, -17 is an inlier, while -18 is an outlier.

5 Submission

The due is **Oct 25th, 23:55 KST**. Pack your package using `npm pack`, and upload the tarball file to iCampus.

- Your code will be graded by an automated program. Double-check the command name and the output format.
- For any cases that are not specified in this document, the right behavior of your program is **undefined**. For example, if a string other than a number is given, you can do anything you want, e.g., printing out an error message or terminating the program.
- Your code will be checked by a copy checker. Keep your code private. This is an assignment, not a real open-source project.
- A report is **NOT** required.
- The three commands we developed in the class, `sum`, `avg`, and `max`, must work as usual even after you implement the commands in this document.

6 Copyright

You will hold the copyright of your work. I will not copy/redistribute/modify your work except for the grading.