COP3275 – Spring 2019

Instructor: Roozbeh Ketabi

Programming Assignment 2

1- You have to upload your codes in both Canvas and the Judge system at https://cop3275.cise.ufl.edu

2- For guide on how to access the Judge, visit Judge Access Page under Pages section in Canvas.

3- The judge will run test cases against your code and assign it a grade. You can have multiple submission for the same problem but you have to choose which one is the final (we will consider your final submission for grading). Canvas is only used for record keeping.

4- When upload on canvas, zip your programs in a zipfile with your ufid as name (nothing more, just 8 digit ufid, for instance 12345678.zip). Name your files (inside the zipped archive) p1.c, p2.c, and so on.

5- For all the problems input is read from standard input (i.e. read using scanf) and must be written to standard output (i.e. printf).

6- **Don’t print extra stuff. Since the assignments are automatically graded, if the output doesn’t match the expected output, you will not get the points. For instance: If the problem is to read a number and return its square, the expected output is a number (i.e. “printf(“%d”,i*i)”). If you print using something like “printf(“square is %d”,i*i);” you will not receive any points.**

7- The assignment is due on 11:59 pm Sunday Feb. 17th 2019. There is a 20% penalty for late submission of one day. No submission is accepted beyond that time.

Problem 1: Parentheses come in pairs (1 pts)

In this problem, we want to find whether a given string is properly parenthesized (by proper we mean every parenthesis has a matching one).
Input format:
You are given input in one line consisting of opening and closing parentheses.

Print 1 if the string is valid and zero otherwise.

Printing 0 or 1 in hopes of getting half of the credit will get you zero points.

Examples:

Input	Output
()()	1
((()))	1
))((	0
(())))	0
(((()())()))	1
(()()))()(())	0
)()(	0

Hint: you can use scanf with %c to scan one character from input at a time.

```
char c;
while(scanf("%c",&c),c!='\n'){
    if(c == ‘(‘)
      …
}
```

Hint:
You can keep track of number of opened parentheses. If at any point you close more than you have opened it is improper. If after you read all the input, there are unmatched open parentheses, it is improper as well.

Problem 2: Stock Market! (3 pts)

In this problem you are given the values of market index through time. You are tasked to compute:

- First day with maximum value. Print both the day and the maximum value.
- First day with minimum value. Print both the day and the minimum value.
- Average value overall.
- Number of days with higher than average value.

Input format:

First comes N the number of days for which we are getting values, in a line.

Each of the following lines would have the value for one day, starting with day 1.

Output format:

In the first line of output, print the day with maximum value and its corresponding maximum value separated with a space.

In the second line, same as above but for minimum.

In third line print the average value to 2 decimal places.

In fourth line print the number of days with a value

Example:

Input	**output**

5
11823.53
11800.92
11788.77
11792.98
11805.04
	1 11823.53
	3 11788.77
	11802.25
	2

Explanation:

The highest value is 11823.53 which occurs on day 1.

The lowest value is 11788.77 which occurs on day 3.

Average (sum divided by count) is 11802.25

There are 2 days with a number greater than 11802.25 namely day 1 with value 11823.53 and day 5 with value 11805.04

Note:

If you get multiple days with the same highest/lowest value, print the first one. You can do so by only updating your temporary _max (_min) only if the current value is strictly bigger (smaller).

Suggestions:

- **Store the input as you read them into an array. Since arrays are indexed from 0 to n-1, corresponding day to ith element is i+1.**
- **Write a function for each task (e.g. a function that finds the maximum element and returns its index in the input array, then you can easily find the value using that index).**

Problem 3: Mind Reader (3 pts)

In this problem you have to write a program that tries to find the number in range [0, 512) that the user has in mind. It works by guessing a number and asking the user whether it is greater than or less than the number they have in mind until it guesses the number. A negative input by user means the number they have in mind is less than the number the computer prints. A positive number means the number they have in mind is greater than the number computer guesses. Zero means the computer got the number right and program exits.

Suggested method of finding the number is through binary search. For that every time you suggest the number right in the middle of the range of values. So, you always start with (0+512)/2 which is 256. If the user input is greater than 0, the next guess would be the middle of range [256,512] which is (256+512)/2 equal to 384. And so on.

Sample:
* assume the user number is 120
Sample interactions:
Computer prints: 256
User input: -1
Computer prints: 128
User input: -1
Computer prints: 64
User input: 1
Computer prints: 96
User input: 1
Computer prints: 112
User input: 1
Computer prints: 120
User input: 0

* **Note that this is the sample interaction, not the output of your program. The output would only consist of the computer guesses. For instance, in this case, the output is:**

```
256
128
64
96
112
120
```