CST 370 – Fall B 2020
Homework 1
Due: 11/03/2020 (Tuesday) (11:55 PM)

How to turn in?

- Submit **three C++ or Java programs** on the iLearn. When you submit your homework programs, don't forget to include "Title", "Abstract", "ID", "Name", and "Date".
- Note that the **due time is 11:55(PM)**. This is the iLearn's timestamp, not your submission time. Since there could be a long delay between your computer and iLearn, you should **submit early**.

1. Write a C++ program (or Java program) called **hw1_1.cpp (**or **hw1_1.java)** that reads input numbers from a user and displays the closest distance between two numbers among all input numbers.

Input format: This is a sample input from a user.

```
5
7
-7
20
8
15
```

The first number (= 5 in the example) indicates that there will be five integer numbers in the input. Then, all numbers from the second line (= 7 in the example) to the last line (15 in the example) are actual numbers. Thus, your program should read them and present the closest distance among five numbers 7, -7, 20, 8, and 15. Because the distance between 7 and 8 is 1, your answer should be 1. Note that the distance between two numbers should be **always positive**. For the problem, you can assume that the actual **numbers are distinct (= no duplication)**.

Sample Run 0: Assume that the user typed the following six lines

```
5
7
-7
20
8
15
```

This is the correct output of your program.

```
Min distance: 1
Two numbers for min distance: 7 and 8
```

Note that **the sequence of two numbers for the minimum distance is important**. Your program has to display the small number first. So, the correct answer is 7 and then 8. If your program displays 8 and then 7, it's not correct.

Sample Run 1: Assume that the user typed the following twelve lines

```
10
-1
10
-15
3
72
20
15
-20
5
30
```

This is the correct output of your program.

```
Min distance: 2
Two numbers for min distance: 3 and 5
```

2. Write a C++ (or Java) program named **hw1_2.cpp** (or **hw1_2.java**) which checks if two strings are anagram or not. In the problem, an anagram is a word formed by rearranging the letters of a different word. For example, the word **EAT** is an anagram of **TEA**. **DEBITCARD** is an anagram of **BADCREDIT**. But LOVE is not an anagram of LIKE.

For this program, you have to use a counter of each character in the input strings to check if the input strings are anagram or not.

Sample Run 0: Assume that the user typed the following input

```
EAT TEA
```

This is the correct output of your program.

```
ANAGRAM
A:1
E:1
T:1
```

Sample Run 1: Assume that the user typed the following input

```
DEBITCARD BADCREDIT
```

This is the correct output of your program.

```
ANAGRAM
A:1
B:1
C:1
D:2
E:1
I:1
R:1
T:1
```

Sample Run 2: Assume that the user typed the following input

```
LOVE LIKE
```

This is the correct output of your program.

```
NO ANAGRAM
```

Sample Run 3: Assume that the user typed the following input

```
COOLVIDEOGAME GIVEACOOLDEMO
```

This is the correct output of your program.

```
ANAGRAM
A:1
C:1
D:1
E:2
G:1
I:1
L:1
M:1
O:3
V:1
```

3. Write a C++ program (or Java program) called **hw1_3.cpp (**or **hw1_3.java)** that reads a set of events and displays the maximum number of events that take place concurrently.

Input format: This is a sample input from a user.

```
5
7 12
10 14
8 11
12 15
16 20
```

The first number (= 5 in the example) indicates that there will be five events in a convention center. All numbers from the second line to the last line are event hours. For example, the first event starts at 7:00(AM) and ends at noon (= 12:00(PM)). The second event starts at 10:00(AM) and ends at 2:00(PM). The last event starts at 4:00(PM) and ends at 8:00(PM).

Sample Run 0: Assume that the user typed the following lines

```
5
7 12
10 14
8 11
12 15
16 20
```

This is the correct output of your program.

```
Max events: 3
```

Note that there will be three events from 10:00(AM) to 11:00(AM). Also, there will be three events at 12:00(PM). So, the maximum number of events that take place concurrently should be 3.

Sample Run 1: Assume that the user typed the following lines

```
10
13 14
1 2
15 16
17 18
7 8
9 10
5 6
11 12
19 20
3 4
```

This is the correct output of your program.

```
Max events: 1
```

Sample Run 2: Assume that the user typed the following lines

```
7
10 14
19 20
12 18
0 24
7 8
7 13
19 20
```

This is the correct output of your program.

```
Max events: 4
```

Sample Run 3: Assume that the user typed the following lines

```
4
10 15
17 18
12 13
16 23
```

This is the correct output of your program.

```
Max events: 2
```

Hint: There are only 24 hours in a day.