PA3

Computer Programming for Engineers
Instructor: Younghoon Kim

Introduction

- **Check PA3 at http://skku.goorm.io/.**

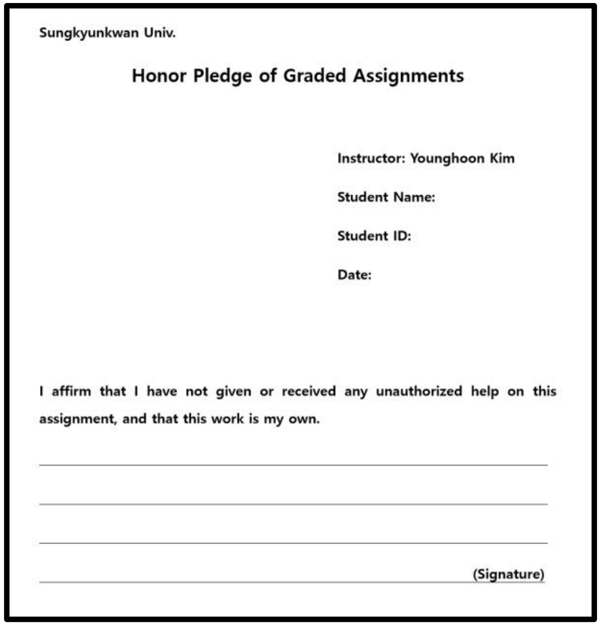
- **Deadline : 2020.12.11 11:59 pm**
- **You can submit PA3 for two more days (~ 2020.12.13 11:59 pm) after the deadline. (25% deduction per day)**

- **Three problems for PA3**
  - Each problem has the same portion.

- **Questions regarding PA3 will not be answered after 2020.12.11 6:00 pm.**

Honor Pledge

- **Please upload the honor pledge to iCampus.**
  - The template of honor pledge will be uploaded on iCampus.
  - Download and print it. (Or you can use your tablet PCs.)
  - Fill in your personal information.
  - **Handwrite** the statement and leave your signature.
  - With the signature of yours, you agree with the disciplinary actions which will be followed after the honor pledge violation.



Sungkyunkwan Univ.

Honor Pledge of Graded Assignments

Instructor: Younghoon Kim

Student Name:

Student ID:

Date:

I affirm that I have not given or received any unauthorized help on this assignment, and that this work is my own.

(Signature)

Info.

- **List of Problems**
  1. Membership
  2. Common Members
  3. Common Members with Duplicates

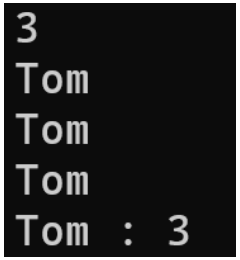
- **Common notice for all problems**
  - You can use any structures supported in C++ STL, such as set, map, vector and priority queue.
  - No special characters in names

Problem 1. Membership

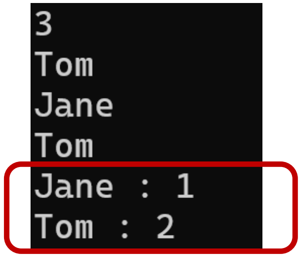
<Description>

- Make a program that counts the number of occurrences of each name and prints out the results in ascending order.
- The details of the program are as follows.
  1. Input the number of names, N.
  2. Input N names.
  3. Calculate the number of occurrences of each name.
  4. Print out the result in ascending order.
     - Output format - Some_name : 1
- Scoring Policy: 33 scores per test case. (Total Score = 100)

<Execution Example>



```
3
Tom
Tom
Tom
Tom : 3
```



```
3
Tom
Jane
Tom
Jane : 1
Tom : 2
```

Ascending order

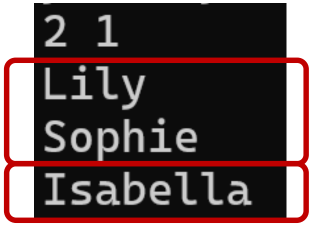
Problem 2. Common Members

<Description>

- Make a program that finds common names from two groups and prints those names in ascending order.
- The details of the program are as follows.
  1. Input two numbers for each group, N1, N2.
  2. Input N1 + N2 names.
     - First N1 names are for the first group and following N2 names are for the second.
     - Each group can have duplicate names.
  3. Find common names and print out them in ascending order.
     - Output format - Some_name
- Scoring Policy: 33 scores per test case. (Total Score = 100)

Problem 2. Common Members

<Execution Example>

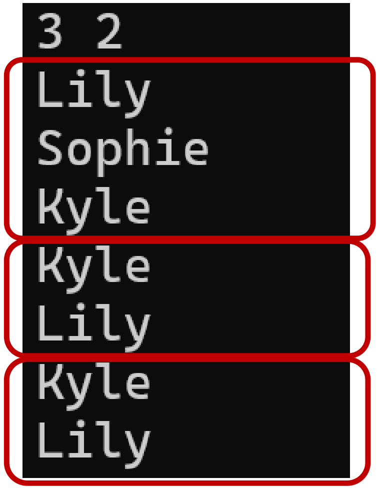


```
2 1
Lily
Sophie
Isabella
```

1st group

2nd group

No common names

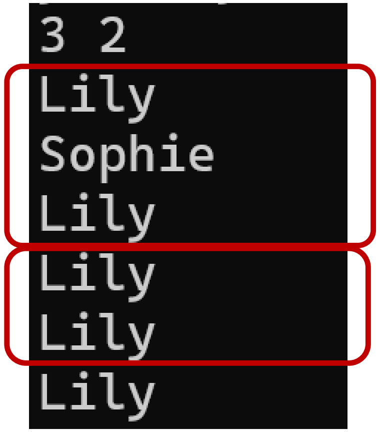


```
3 2
Lily
Sophie
Kyle
Kyle
Lily
Kyle
Lily
```

1st group

2nd group

Ascending order



```
3 2
Lily
Sophie
Lily
Lily
Lily
Lily
```

1st group

2nd group

Only one common name

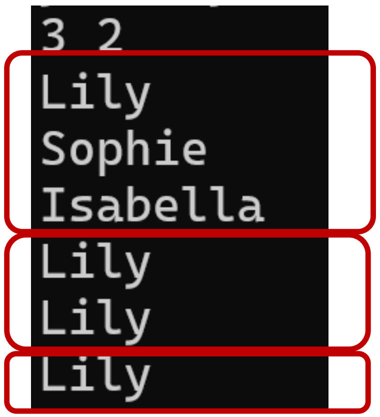
Problem 3. Common Members with Duplicates

<Description>

- Make a program that finds common names including duplicates from two groups and prints those names in ascending order.
- The details of the program are as follows.
  1. Input two numbers for each group, N1, N2.
  2. Input N1 + N2 names.
     - First N1 names are for the first group and following N2 names are for the second.
     - Each group can have duplicate names.
  3. Find common names including duplicates and print out them in ascending order.
     - Output format - Some_name
- Scoring Policy: 33 scores per test case. (Total Score = 100)

Problem 3. Common Members with Duplicates

<Execution Example>

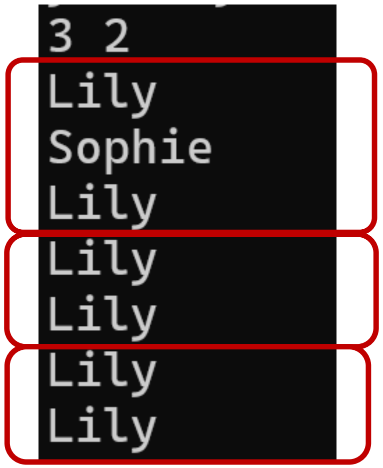


```
3 2
Lily
Sophie
Isabella
Lily
Lily
Lily
```

1st group

2nd group

Only one common name



```
3 2
Lily
Sophie
Lily
Lily
Lily
Lily
Lily
```

1st group

2nd group

We have two "Lily"s including duplicates.
You have to print them all.