Introduction to Computers and Programming LAB-13 2016/01/06

✧ Your input/output must follow our format in examples

✧ Please sign in the computer with your own account of CS Computer Center if you have it, and save your codes in your own space in case of something bad happens on your computer.

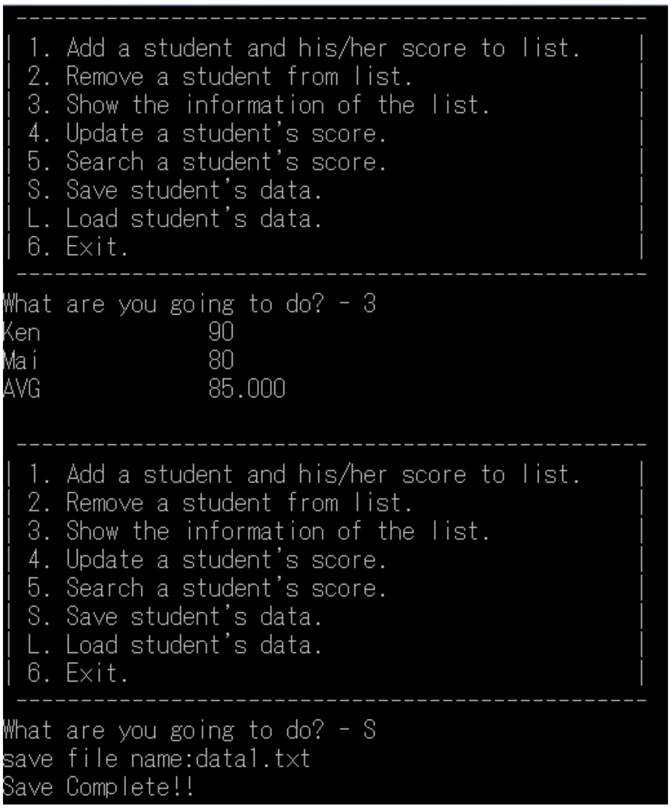
✧ If you cannot finish it in time, you should demo your lab work at next lab time.
(The deadline is 21:30).

1. SOS from TAs_File

You have already given TAs a hand twice, but TAs have a trouble again!! We want you to add "Save" & "Load" function to this students' database program. Follow by this:

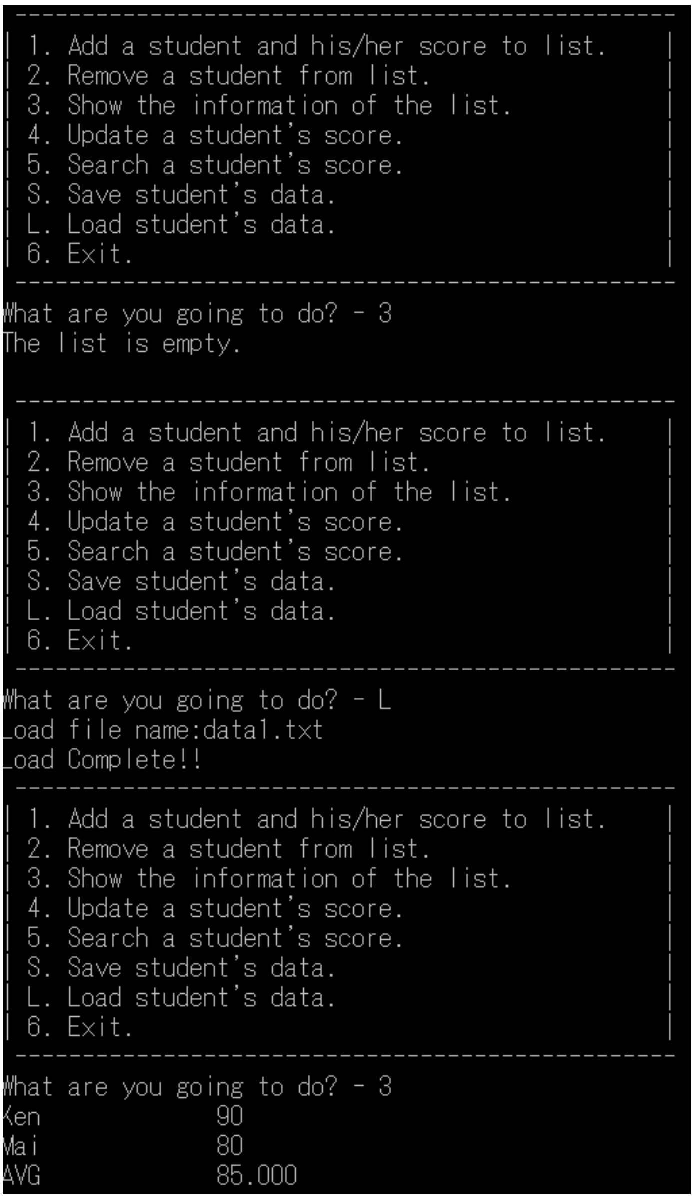
I. Implement save & load function to save (or load) a list of students' database. (The file format without any restricted)

II. The length of file path isn't greater than 100.



```
 -----------------------------------------------
| 1. Add a student and his/her score to list.   |
| 2. Remove a student from list.                |
| 3. Show the information of the list.          |
| 4. Update a student's score.                  |
| 5. Search a student's score.                  |
| S. Save student's data.                       |
| L. Load student's data.                       |
| 6. Exit.                                      |
 -----------------------------------------------
What are you going to do? - 3
Ken            90
Mai            80
AVG            85.000

 -----------------------------------------------
| 1. Add a student and his/her score to list.   |
| 2. Remove a student from list.                |
| 3. Show the information of the list.          |
| 4. Update a student's score.                  |
| 5. Search a student's score.                  |
| S. Save student's data.                       |
| L. Load student's data.                       |
| 6. Exit.                                      |
 -----------------------------------------------
What are you going to do? - S
save file name:data1.txt
Save Complete!!
```



```
 -----------------------------------------------
| 1. Add a student and his/her score to list.   |
| 2. Remove a student from list.                |
| 3. Show the information of the list.          |
| 4. Update a student's score.                  |
| 5. Search a student's score.                  |
| S. Save student's data.                       |
| L. Load student's data.                       |
| 6. Exit.                                      |
 -----------------------------------------------
What are you going to do? - 3
The list is empty.

 -----------------------------------------------
| 1. Add a student and his/her score to list.   |
| 2. Remove a student from list.                |
| 3. Show the information of the list.          |
| 4. Update a student's score.                  |
| 5. Search a student's score.                  |
| S. Save student's data.                       |
| L. Load student's data.                       |
| 6. Exit.                                      |
 -----------------------------------------------
What are you going to do? - L
Load file name:data1.txt
Load Complete!!
 -----------------------------------------------
| 1. Add a student and his/her score to list.   |
| 2. Remove a student from list.                |
| 3. Show the information of the list.          |
| 4. Update a student's score.                  |
| 5. Search a student's score.                  |
| S. Save student's data.                       |
| L. Load student's data.                       |
| 6. Exit.                                      |
 -----------------------------------------------
What are you going to do? - 3
Ken            90
Mai            80
AVG            85.000
```

2. Find different words

Please write to calculate how many different words in given files. (Please Ignore case)

(There will no more than 1000 different words, and max length of word is 100)

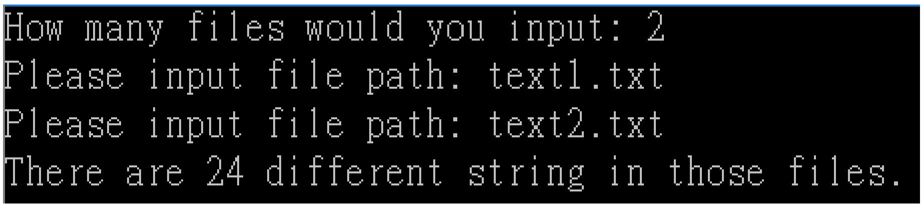
Input:

1. How many files
2. Path of each file

Output:

How many different words in those files

Example:



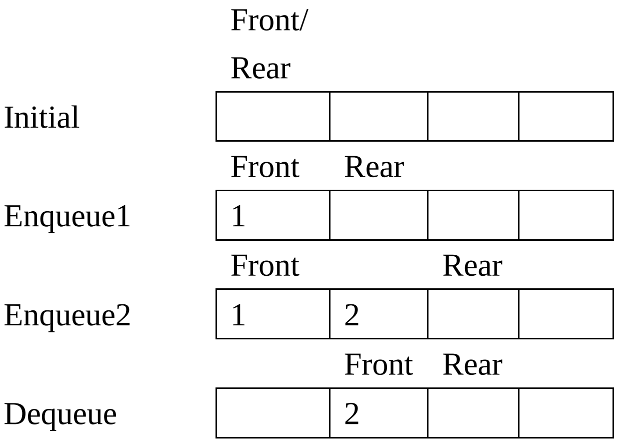
```
How many files would you input: 2
Please input file path: text1.txt
Please input file path: text2.txt
There are 24 different string in those files.
```

3. Simple Queue

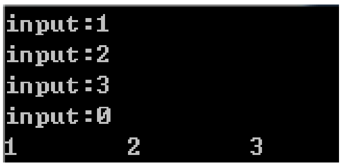
In computer science, a queue is a particular kind of abstract data type or collection in which the entities in the collection are kept in order, known as “ Enqueue ” , and removal of entities from the front terminal position, known as “ Dequeue ” .

Now your program should be like a queue that max size 5. Here is a sample for how a queue works and some functions you need to implement:

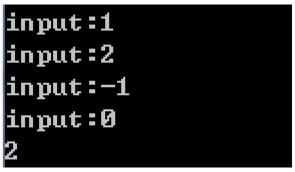
Hint: A queue usually has two index (Font, Rear) to make good use of the memory space.



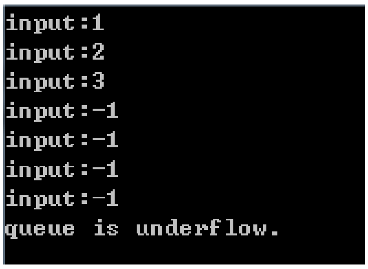
1. “0”means stop the program, print all the element inside after this command.
2. “-1”means dequeue, delete the element from the front.
3. Other integer will be taken as the input data.
4. While the size of this queue is fixed, when the queue is full or underflow you need to show messages then do the same thing as “0”



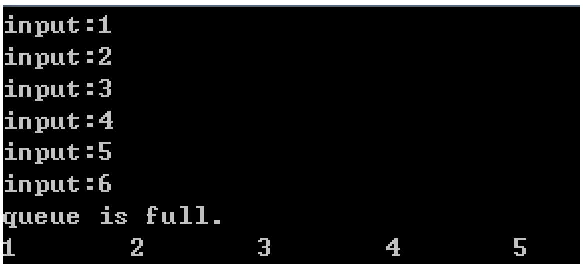
```
input:1
input:2
input:3
input:0
1       2       3
```



```
input:1
input:2
input:-1
input:0
2
```



```
input:1
input:2
input:3
input:-1
input:-1
input:-1
input:-1
queue is underflow.
```



```
input:1
input:2
input:3
input:4
input:5
input:6
queue is full.
1       2       3       4       5
```

4. The good, the bad and the ugly

Given an unsigned integer ≥ 1, determine if it is good, bad, and ugly. The definition of a good/bad/ugly number is described as following:

(1) Good number

An unsigned integer is good if **all the numbers formed by 3 adjacent digits of it are distinct**. That is, the number $d_kd_{k-1}d_{k-2}\cdots\ d_2d_1d_0$ is good if the numbers $d_kd_{k-1}d_{k-2},\ d_{k-1}d_{k-2}d_{k-3,}\cdots, d_2d_1d_0$ are all distinct.

Note that an unsigned integer is good if it is formed by 3 digits or less. Examples:

A good number	1	
	1234567890	123, 234, 345, 456, 567, 678, 789, 890 are all distinct
Not a Good number	1093500000	000 occurs thrice
	3456565657	565 occurs thrice and 656 occurs twice

(2) Bad number

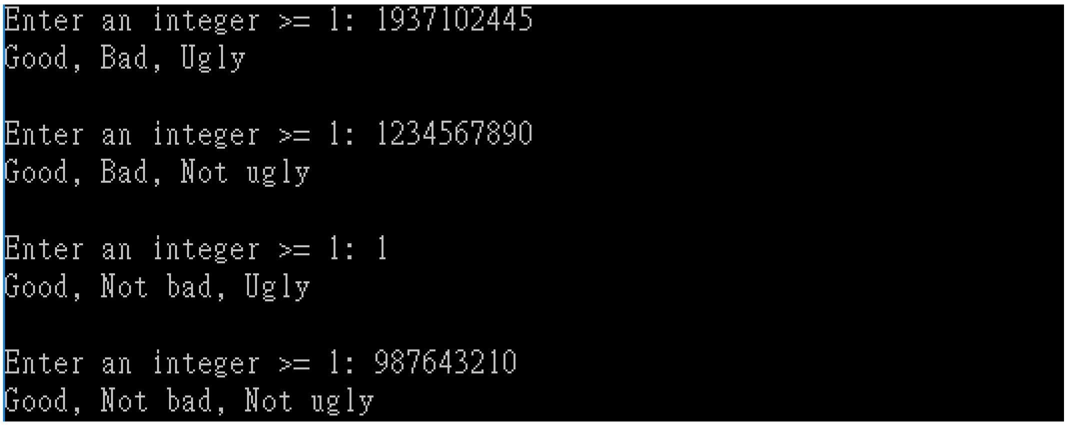
An unsigned integer is bad if it **contains the number 1, 3, and 5**. Note that a bad unsigned integer must have at least 3 digits. Examples:

A bad number	135135	
	1234567890	
Not a bad number	1	3 and 5 don’t occur in it
	3456565657	1 doesn’t occur in it

(3) Ugly number

An unsigned integer ≥ 1 is ugly if its **only prime factors are 2, 3 or 5**. By convention, 1 is an ugly number. Examples:

A ugly number	1937102445	1937102445 = 3^18 * 5^1
	2361960000	2361960000 = 2^6 * 3^10 * 5^4
Not a ugly number	135135	135135= 3^3 * 5^1 * 7^1 * 11^1 * 13^1
	1234567890	1234567890 =2^1 * 3^2 * 5^1 * 3607^1 * 3803^1



```
Enter an integer >= 1: 1937102445
Good, Bad, Ugly

Enter an integer >= 1: 1234567890
Good, Bad, Not ugly

Enter an integer >= 1: 1
Good, Not bad, Ugly

Enter an integer >= 1: 987643210
Good, Not bad, Not ugly
```