

Lab Assignment-1
Summer Semester UCS-301
[5 marks]
Last date of submission: 1 July 2022

Question 1: In a library a librarian is stacking the books using their numeric ids. He may push or pop a book from top and want to know which book ids are currently in the stack. Make a class-based menu driven program to take size of stack in 1st line and then either of options – pop, push or stop. If pop or stop option is given no numeric value follows but if push is given a numeric value to be pushed is given.

Sample Input:

5 push 3 push 1 push 9 push 11 pop push 4 push 7 pop stop

Output:

Welcome to stacks!

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Element popped: 11

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Element popped: 7

Give one of options: pop, push, stop

Stack elements are:

|4|

|9|

|1|

|3|

Question 2: In an examination a teacher evaluates all answer sheets given to him in alphabetical order and keeps on piling checked notebooks over previous checked notebooks. Like this he reverses all the notebooks. Now he wants to distribute to students in alphabetical order, use 2 stacks to help him make the notebooks appear in alphabetical order. He may also pop few answer sheets from top at any point of time to consider later and they will not be shown in output. (Assume input already in alphabetical order). (1st line of input the size of stack and then followed by push, pop or stop, only push followed by first name of student).

Sample Input:

15 push Abhay push Amit push Bharat push Jay push Kartik push Mohit pop push Rohit push Saurabh stop

Output:

Welcome to stacks!

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Element popped: Mohit

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Student notebooks are:

|Abhay|

|Amit|

|Bharat|

|Jay|

|Kartik|

|Rohit|

|Saurabh|

Question 3: In a exam a stack of notebooks is given in a circularly sorted fashion and we want to arrange them in stack with correct sorting with least element on top. We have to use stack only and no sorting or other program. Assume no pop occurs only push occurs. (1st line of input the size of stack and then followed by push, pop or stop, only push followed by book id)

Sample Input:

15 push 4 push 5 push 6 push 1 push 2 push 3 stop

Output:

Welcome to stacks!

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Give one of options: pop, push, stop

Stack elements are:

|1|

|2|

|3|

|4|

|5|

|6|

Question 4: SIMPLE-QUEUE

Previously, you have implemented the Stack data structure that follows LIFO structure. There is another data structure that is used heavily in day-to-day life and in computers named Queue. Queue is a simple data structure to store the elements in the first-in-first-out (FIFO) order. The new elements are added at the end of the queue and items are deleted from the front of the queue. Two variables (namely front and rear) are used to keep track the record of the current state of the queue. The common operations performed onto the queue are the insertion of new element, deletion of the existing element, and traversing (displaying) all the elements of the queue. Your task is to create a menu driven program to implement the functionality of the queue using array. You are also required to count and display the number of insertion and deletions performed. In order to implement the functionality, do the following:

- Create a class named SIMPLE_QUEUE.
- Add a constructor to initialize the front and rear variables to -1.
- Add method named enqueue(), dequeue() to add to and delete from the queue.
- Add another method to display the contents of the queue.
- Your program should display appropriate message when there no space to add a new element or if there is nothing to delete.

Sample input

```
1 45 1 90 1 88 2 3 5
```

Explanation:

Where the first number refer to the type of operation we want to perform (insertion=1, deletion=2, display=3, stop=4). In the above line first number is 1, it means the value 45 will be inserted into the queue. Similar for value 90 and 88. Basically there are three insertions in the queue. Followed by one deletion, so number first inserted number will be deleted from the queue. And finally 3 is the code to display all the element of the queue, and finally values other than 1, 2 and 3 will lead to the termination of the program.

Sample output

The contents of the queue are:

Q[0]=90 Q[1]=88

Inserted=3, Deleted=1

Test case 1

Input

```
1 45 1 56 1 -75 2 1 -20 2 2 3 10
```

Output

The contents of the queue are:

Q[2]=-20

Inserted=4, Deleted=3

Test case 2

Input

1 45 1 -25 2 2 3 20

Output

The contents of the queue are:

Queue is empty, there is nothing to display.

Inserted=2, Deleted=2

Test case 3

Input

Output

Inserted=0, Deleted=0

Test case 4

Input

3 6

Output

The contents of the queue are:

Queue is empty, there is nothing to display.

Inserted=0, Deleted=0

Question 5: Movie-Tracker

In question 1, you have implemented the basic queue operations; insert, delete, and traverse. Now, here your task is to count the total number of males, females, and kids entered into the INOX Cinema so that the owner can decide the discount policy accordingly. The male, female, and kids follow a single line (queue) to enter into the theatre. The security person is not able to keep track the records: how many males, female and kids entered into the theatre? Instead, he just maintains a register in which he writes 'M', 'F', 'K' whenever any male, female, or kid entered into the theatre. The ticket booking person has the total number of the tickets sold for the day and at the end of the day writes the total in front of the entries made by the first security guard. One typical entry of the register looks like:

5 M F F K M.

Sometimes it is also possible that the first security guard may enter wrong entries into the register (other than M, F, K).

In order to solve the problem, you are required to do the following

- Create a class named "QueueCounter" with basic insert, delete and traverse function.
- Create two variables named **front** and **rear** to mark the status of the queue. Also, add another variable named **counter** to that count the total number of male, female or kids for each.

- Create a separated functions to increment and display the **count** variable.
- In the main function you need to create three instances of the class “QueueCounter” as “males”, “females”, and “kids” and add teach incoming entry to the respective queue instance only.

Sample input

7 M F K K F F M

Sample output

Total Males are:2

Total Females are:3

Total Kids are:2

Test case 1**Input**

16 M M F K K F F M M K F M F F K K

Output

Total Males are:5

Total Females are:6

Total Kids are:5

Test case 2**Input**

9 F K K F F K F F F

Output

Total Males are:0

Total Females are:6

Total Kids are:3

Test case 3**Input**

5 M F K F F K F F

Output

Total Males are:1

Total Females are:3

Total Kids are:1