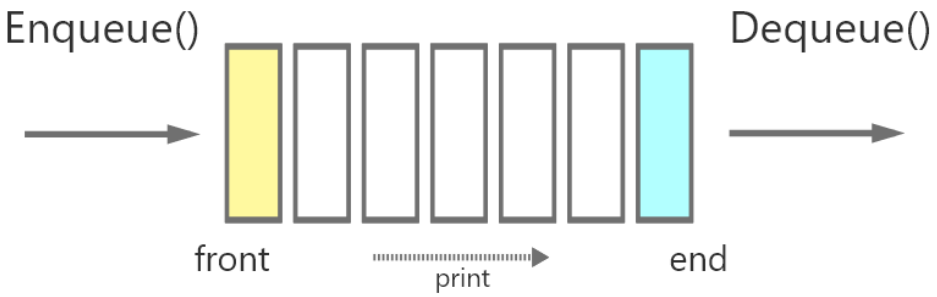
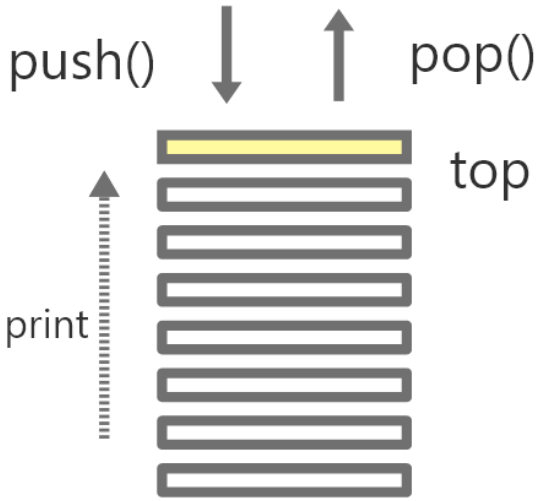
Lab9. Stack and Queue

---

queues are a type of container adaptors, specifically designed to operate in a FIFO context (first-in first-out), where elements are inserted into the **end** of the container and extracted from the **front**.



stacks are a type of container adaptors, specifically designed to operate in a LIFO context (last-in first-out), where elements are inserted and extracted only from top(**front**) of the container.



You should build the project on your own.
You need to implement the read file first. And then implement the following.
There are 6 functions to be completed in your project.
In stack class

- void push(int num) : push an integer number into the top of the stack
- void pop() : pop out the integer number at the top of the stack, do nothing if the stack is empty
- void print() : print all the elements in the stack

In queue class

- void enqueue(int num) : push an integer number into the end of the queue

- void dequeue() : remove the integer number at the front of the queue, do nothing if the queue is empty
- void print() : print all the elements in the queue

You must use Linked-list to implement these functions.

Input Format

The first line of the input file means the number of test cases.
For each test case, the first character means which container adapter (Stack or Queue) that you need to use **Linked-list** to implement. The following number means the number of commands in the next line.

Container adaptor :
S : Stack
Q : Queue

Commands :
push 2 : void push(2)
pop : void pop()
print : void print()

en 2 : void enqueue(2)
de : void dequeue()
print : void print()

Notice that print a blank line if the stack or queue is empty in executing the print command.

Output Format

See more detail from Sample Input and Sample Output.
Your output format should be the same as the sample output, so observe sample output carefully.

Sample Input

3
S 8
push 2 push 3 push 5 push 7 pop print push 6 print
Q 8
en 2 en 3 en 5 en 7 de print en 6 print
S 4
pop print push 1 print

Sample Output

The values in the stack : 2 3 5

The values in the stack : 2 3 5 6

The values in the queue : 7 5 3
The values in the queue : 6 7 5 3

The values in the stack :
The values in the stack : 1