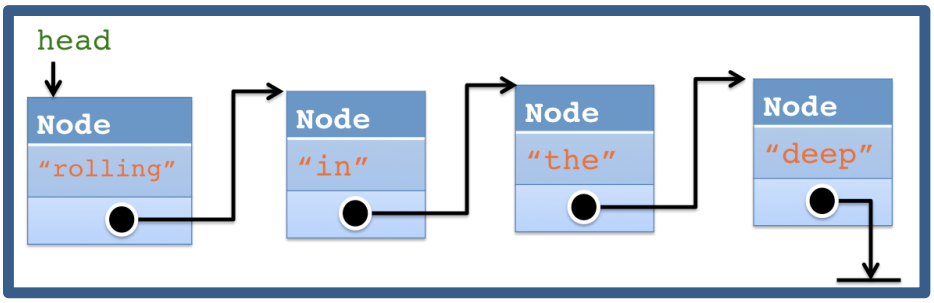
Introduction to Programming

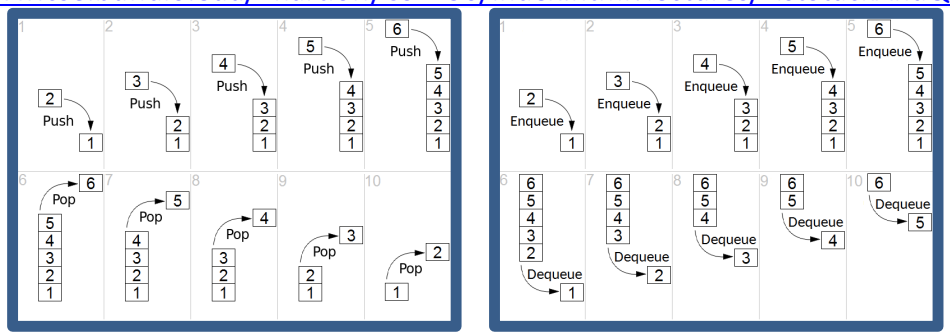
(CS200)

Shafay Shamail

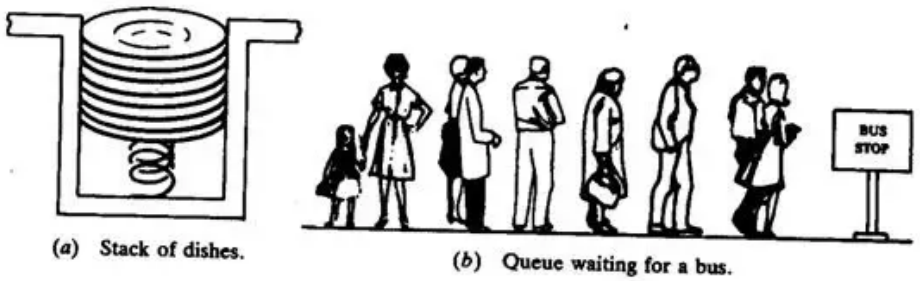
Lists, Stacks, Queues



https://www.cse.buffalo.edu/~hartloff/CSE250/index.html?lectures/ListStackAndQueue.html



https://commons.wikimedia.org/wiki/File:Lifo_stack.png
https://commons.wikimedia.org/wiki/File:Fifo_queue.png



(a) Stack of dishes. (b) Queue waiting for a bus.

https://www.quora.com/What-are-some-real-world-applications-of-a-queue-data-structure-1

CS 200 Lab 06 Spring 2018

Lab Guidelines

1. Make sure you get your work graded before the lab time ends.
2. You put all your work onto the LMS folder designated for the lab (i.e. "Lab05") before the time of the lab ends.
3. Talking to each other is NOT permitted. If you have a question, ask the lab assistants.
4. The object is not simply to get the job done, but to get it done in the way that is asked for in the lab.
5. Any cheating case will be reported to Disciplinary Committee without any delay.

NOTE: **Define a class interface separately and its methods separately. Do not write inline code.**

Task 1 is for those whose roll number is odd and Task 2 is for those whose roll number is even.

Marks: Name: _____ Roll #: _____

Task 1	A	B	C					Total
	40	30	30					100

Task 2	A	B-I	B-II					Total
	40	30	30					100

Let's Begin……

Total marks Obtained

/100

CS 200 Lab 06 Spring 2018

Task 1(A) - STACKS: (40)

Write a class **Stack** that will have following member functions: 5

- push() //inserts an element at the end of the stack 5
- pop() // removes the top element 5
- top() // returns the top element in the stack 5
- size() // returns the size of the stack 5
- isEmpty() //checks if stack is empty 5
- isFull() //checks if stack is full 5

Note:

- ⇨ Limit array size to 25.
- ⇨ You must implement it using arrays that store strings.
- ⇨ Use array indexing where ever required.
- ⇨ Make appropriate constructor(s) and destructor as well. 5

STOP AND SHOW YOUR WORK TO THE TA

Task 1(B) – Balance Parentheses: (30)

Write a program that reads a string of parentheses, square brackets, and curly braces from standard input and uses a stack to determine whether they are properly balanced.

For example:

Input: [[( )]] => **returns True**

Input: [{ { (}) }] => **returns False**

Note: You must use the stack developed in Task1(A).

STOP AND SHOW YOUR WORK TO THE TA

Task 1(C) – Expression Evaluation: (30)

Write a program that takes an Arithmetic Expression and evaluates it using stacks.

For example, you will be given an expression like:

(3 + 7) * 6 + (5 % 2 2) + (6 -100)

Note:

- The expression will be given as a string and you have to convert this string into substrings so that each substring is either an operator, a value or a parenthesis.
- You can implement it like this:
  - Push numbers in stack.
  - When an operator is encountered, pop one number and apply the operation on that number and the number after the operator.
  - Evaluate it and push the result back on to stack.
  - In case of parenthesis, … (figure out yourself!)
- You can convert strings to integers as follow:

```
String s = "200";
int x = atoi ( s.c_str() );
```

Add this header file "cstdlib" as well.

STOP AND SHOW YOUR WORK TO THE TA

CS 200 Lab 06 Spring 2018

Task 2(A): QUEUE (40)

Write a class **Queue** that will have following member functions: 5

- enqueue() //adds an element in the queue 5
- dequeue() //deletes an element from the queue 5
- isEmpty() //checks if the queue is empty 5
- isFull() //checks if the queue if full 5
- count() //returns number of elements present in the queue 5
- max() //returns the largest number in the queue 5

⇨ Limit queue size to 25.
⇨ Each element in the queue will be an integer.
⇨ You must implement it using arrays.
⇨ Array indexing is not allowed. Only pointer arithmetic can be used.
⇨ Make constructor(s) and destructor as well. 5

Task 2(B): QUEUE (60)

In the LUMS CS-Smart lab, there is only one printer and sometimes students must wait for a long time to get a single page of output. But since some jobs are more important than others, IST added a simple priority system for print job. Now, each job is assigned a priority between 1 and 9 (with 1 being the lowest).

The printer operates as follow:

- The first job is taken from the queue.
- If any job in the remaining queue has higher priority than this one, the current job is enqueue'd back in the queue and a new job is dequeue'd.
- Otherwise, job is printed.

The problem with this approach is that it has become quite tricky to determine when your print job will be completed.

2(B)-I Your task is to write a function that takes a queue and current position of your job in the queue and returns the time you must wait until your print job is completed. Assume that each job takes exactly 1 minute to complete. **30**

STOP AND SHOW YOUR WORK TO THE TA

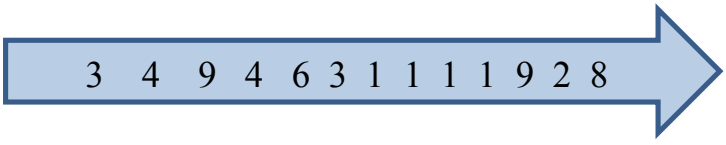
Write a function that prints the items in the queue according to the priority assigned to them. **30**

STOP AND SHOW YOUR WORK TO THE TA

Note: Jobs in the queue are not stored in order of priority. They are stored in order of their arrival.

For Example:

Let's assume our queue is like this and your job is now at position 11.



Pos. 13 12 11 10 9 8 7 6 5 4 3 2 1

It will be 2nd job that will be printed. (Figure this out yourself!)

STOP AND SHOW YOUR WORK TO THE TA

CS 200 Lab 06 Spring 2018

Zip your tasks into one folder with format:
YourRollNo-Lab06
example "**2001001-Lab06**" and upload on LMS before the tab is closed. You will not be given extra time.