

Assignment 3 (Stack and Queue)



Session 18

Fall 2019

Data Structures and Algorithms

Department of Computer Science

University of Engineering and Technology, Lahore (KSK Campus)

Dead Line: Fri 18th Oct till 11:55 pm**Problem 1:** [10]

Write a code to implement the circular Queue using array which will contain generic data item.
(Hint: use template)

Problem 2: [25]

Expression is a sequence of operand and operators. For example, A and B are two operands and '+' is a operator. Then '**A+B**' is an expression. And expression can be written in three different forms with respect to the position of operator.

Form 1: A+B (operator is in between operands) This type of notation is called **Infix**

Form 2: AB+ (operator is after operands) This type of notation is called **postfix**

Form 3: +AB (operator is before operands) This type of notation is called **prefix**

Part (A) [10]

Write a code using stack to convert the Infix expressions entered by the user into PostFix and PreFix notation. The operator appear in the infix expression can be +, -, /, * and %. Write the following functions.

string **PostFix**(Stack st, string str)

string **PreFix**(Stack st, string str)

Infix Notation	Postfix Notation	Prefix Notation
2+5-3/8	2 5 + 3 8 / -	+2-5/38
7/3*4+5	7 3 / 4 * 5 +	+/7*345
(a*(b-c)/d)	abc-*d/	*a/-bcd

Part (B) [15]

Write a code using stack to evaluate the Infix expressions entered by the user. The operator appear in the infix expression can be +, -, /, * and %. Write the following functions.

int **Evaluate**(Stack st , String Exp)

Infix Notation	Solution
2+5-16/8	5
6/3*4+5	13

Problem 3:**[15]**

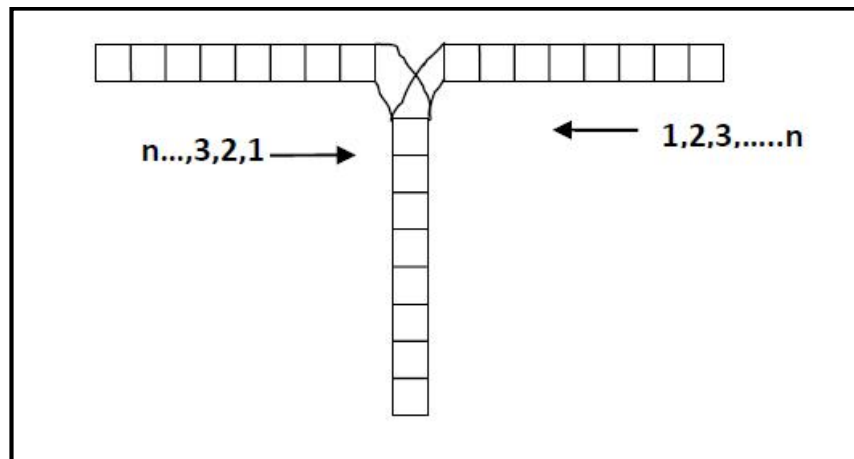
We want to add two numbers. The number of digits that a number contain can be up-to 60 digits. Write a code using stacks to add two numbers.

string **Sum**(String Num1 , String Num2)

<p>Sample input 1 Enter First String: 9568 Enter Second String: 3215</p> <p>Sample Output: Sum of two Numbers: 12783</p>	<p>Sample input 2 Enter First String: 782 Enter Second String: 39</p> <p>Sample Output: Sum of two Numbers: 821</p>
--	---

Problem 4:**[20]**

Consider the railroad switching network given in following figure. Cars can come from either direction from left to right and from right to left. Now both sides of cars have to go out from other direction, i.e. as they are put in a queue. Unfortunately, we have only one track, one car entered from left side can only leave if there is no further car from right to left road. If there are cars we move them in a vertical track and then from left a car move easily to right side. Your task is to design a Data Structure and helping classes which can move cars from either side in this railway road in an efficient manner.



How Simulation Works just understand the given scenario

Enter cars on the Left Road:

10

20

30

Enter cars on the Right Road:

40

50

60

Display Traffic on Left Road: 10 20 30

Display Traffic on Right Road: 40 50 60

Move Car to Right:

Sorry Right Road is busy. Let just move the Right Traffic. Now Right Road is Free.

Display Traffic on Right Road: Empty Road

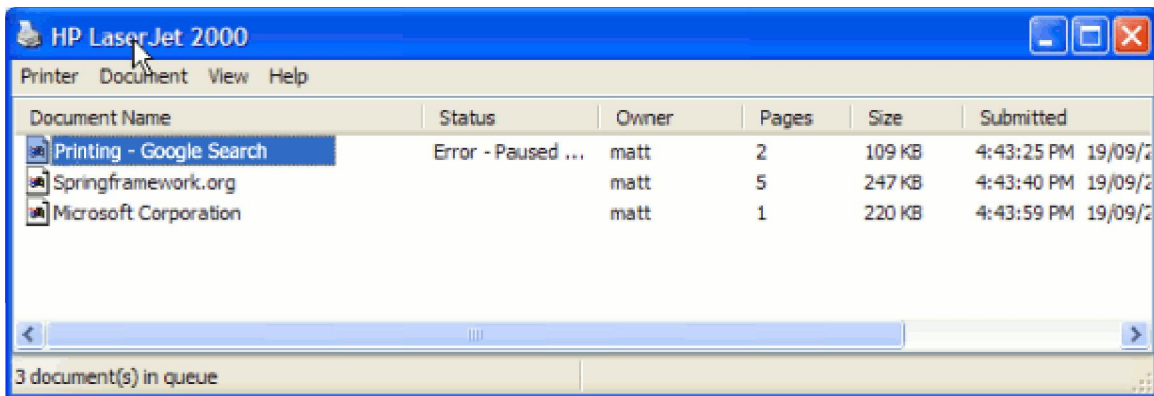
Now car 10 easily Move to Right Road.

Display Traffic on Left Road: 20 30

Display Traffic on Right Road: 40 50 60

Problem 5:**[30]**

You have to simulate the printer queue (printer's printing jobs) by using Queue class. Queue follows FIFO (first in first out) rule. Recall that as students send printing tasks to the shared printer, the tasks are placed in a queue to be processed in a first-come first-served manner. You can understand it by the following picture.



You have to work on PrintingJob where the printer is capable of storing and processing the input jobs. Each job has an associated timeOfArrival, and number of pages in the input file. The printer prints 6 pages in a minute. For example, consider if printing job is sent to a printer at 12:05:30 time and it is composed of 60 pages. It will take 10 minutes to complete this job, therefore it will start at 12:05:30, end at 12:15:30. During this one more job is arrived at 12:07:00 time, which is composed of 12 pages, then this job has to wait in the queue, and would be assigned the starting time as 12:15:30 and would take another 2 minutes to finish, thus would be marked with the end time 12:17:30.

You have to maintain queue and have to do the following:

- 1) calculate the total time for which all printing jobs will be done.

- 2) Furnish each job with its start time and end time (i.e. when would the printer start printing a job, and when would it end printing it.)

Create a class **PrinterQueue**, create a function **processJobs** that takes input array of **PrintingJobs** where each job is represented with Sender name, submission_Time, start_Time, end_Time, Pages. At the time of submission each job is represented by sender name, numberOfPages and submissionTime. The function processJobs manipulates all jobs using a queue and assigns each job the startTime and endTime. It also computes the total time taken to print all jobs in the input array.