# CS593: Data structure and Database Lab
## Take Home Assignment - 4 (10 Questions, 100 Points)

**Submission Dead Line: 18-Sept-2021 23:59 Hours**

**Important Instructions**

- **I.** Hard coding is not allowed for any questions. In case of any hard coding your submission shall not be considered for evaluations. All the example and input case are provided as explanations. So that questions will be easy to understand.
- **II.** In case of copy and plagiarism your submission shall not be considered for evaluations. Negative of maximum marks for assignments 4 will be awarded as penalty.
- **III.** Do not use any unnecessary library. Create your own data structure wherever required
- **IV.** Marks will be awarded if and only if your program pass all test case. If your program fail any test case no marks will awarded for that particular questions.

1) It is possible to keep two stacks in a single array, if one grows from position 1 of the array, and the other grows from the last position. Write a procedure PUSH(x, S) that pushes element x onto stack S, where S is one or the other of these two stacks. Include all necessary error checks in your procedure.

2) We can store k stacks in a single array if we use the data structure suggested in Fig. 2.27, for the case k = 3. We push and pop from each stack as suggested in connection discussion in our class . However, if pushing onto stack i causes TOP(i) to equal BOTTOM(i-1), we first move all the stacks so that there is an appropriate size gap between each adjacent pair of stacks. For example, we might make the gaps above all stacks equal, or we might make the gap above stack i proportional to the current size of stack i (on the theory that larger stacks are likely to grow sooner, and we want to postpone as long as possible the next reorganization).
   a) On the assumption that there is a procedure reorganize to call when stacks collide, write code for the five stack operations.
   b) On the assumption that there is a procedure makenewtops that computes newtop[i], the "appropriate" position for the top of stack i, for $1 \le i \le k$, write the procedure reorganize. Hint. Note that stack i could move up or down, and it is necessary to move stack i before stack j if the new position of stack j overlaps the old position of stack i. Consider stacks 1, 2 , . . . , k in order, but keep a stack of "goals," each goal being to move a particular stack. If on considering stack i, we can move it safely, do so, and then reconsider the stack whose number is on top of the goal stack. If we cannot safely move stack i, push i onto the goal stack.
   c) Implement makenewtops in such a way that space above each stack is proportional to the current size of that stack.

3) Write a program to implement
   a) Stack using two Queues.
   b) Queue using two Stack
4) Write a program using stack to remove recursion from following functions.

```
Comb(n,m)
{
 If ((n==1) or (m==0) or (m==n)) then
```

```
        return 1
    else
        return (comb(n-1 , m) + comb(n-1, m-1))
```

5) A dequeue (double-ended queue) is a list from which elements can be inserted or deleted at either end. Develop array and Linked list implementations for a dequeue.

6) Write a program which takes another c program as input and print “yes” if input c program has parenthesis imbalance error print no otherwise. Print number of “(“ or “)” should be added to make it balanced.

7) In Unix-style file-system, a dot(.) represents the current directory, two dots(..) represent the previous directory by a level, and any multiple consecutive slashes (i.e., '//') are treated as a single slash '/'. Given a string as a path, return the simplified canonical path.

   The path starts with a single slash '/'.
   Any two directories are separated by a single slash '/'.
   The path does not end with a trailing '/'.
   The path only contains the directories on the path from the root directory to the target file or directory (i.e., no period '.' or double period '..')

   Example: path = "/Desktop/"
   Output: "/Desktop"
   Note: There is no trailing slash after directory name

   Example: path = "/root/Desktop/../"
   Output: "/root"

   Example: path = "/a/./b/../../c/"
   Output: "/c"

8) Write a program to implement First come First out (FIFO) page replacement algorithm.

9) Write a program to convert
   A. Infix expression to postfix expression
   B. Infix expression to postfix expression
   C. Prefix expression to postfix expression

10. Write a program to implement
    a) Round Robin scheduling algorithm
    b) Shortest job remaining First scheduling algorithm

## Submission instruction

**File Naming Convention:** Create a directory with your roll number. Inside this directory, place all the programs and input files. Prefix the file name with your roll number followed by “_” followed by question number followed by “.c”. Example: 194161000_q1. c.

**README.txt** Write a short note on sequence of steps involved to run your programs. Include what is the input for the program (with an example) and what will be the output from the program (with an example).

**tar gzip** Create (roll number). tar.gz file using the above directory. This directory must contain the above program.

**Submission** Email the above tar gzip file to the CS593 head TA **sujitkumar@iitg.ac.in** as per the above given dead line.

**Submission** does not follow above file naming convention and instruction shall not be evaluated.

**Copying** You should avoid indulging in copying. Every submission will be subject to software similarity using the **tool Measure of Software Similarity available at https://theory.stanford.edu/~aiken/moss/.** Do not copy code from any source including coding website. Two submissions having similarity score equal to or more than 10.0% will be declared copied. If you are found involved in copying act, your name will be referred to disciplinary committee. Therefore, you are requested to place individual efforts and avoid copying.