UCS 1312 Data Structures Lab

A4: Applications of Stacks and Queues

-- Dr. R. Kanchana

Best Practices to be adapted

Modular design and coding using versions

Improve readability of code by making the program self-explanatory, giving meaningful names to your variables and functions

Avoiding global variables

^^^^^^

Write algorithms for applications in question 1 and 2 and trace them with an example. Inspect the steps using the diagrammatic representation of the Stack / Queue.

- 1. Create an ADT for a Stack of strings (string.h) implemented using arrays.
- (CO1, K3)

a) Add the following operations:

push, pop, isFull, isEmpty, getTop

- b) Write an application for the following that uses the stack (a4stack.c)
 - 1. Given an arithmetic expression, convert to postfix and evaluate it.
 - II. Given an expression with two types of parenthesis ([]), check whether the parenthesis are balanced.
- c) Demonstrate the stack operations and applications with suitable test cases
- 2. Create an ADT for Queue of integers implemented using a circular array (Queue.h) (CO1, K3)
 - a) Add the following operations:
 - enqueue. dequeue, isFull, isEmpty, getRear,getFront
 - b) Write an application for the following that uses the Queue (a4q.c)

Consider a Printer Spooler for a network printer — jobs submitted to a printer form a queue for that printer and the job are printed in spooled order. The interface for the printer spooler has the operations: *spool, print, list*. Implement them using the Queue ADT.

- 1. spool(q, jobID) adds a job to q.
- 2. print(q) removes the job to be printed from q.
- 3. list(q) lists the jobs in q in the spooled order.
- c) Demonstrate the Queue operations and applications with suitable test cases
