ACADEMY OF TECHNOLOGY
Lab Assignment (Day 1)
Paper name: Design and Analysis of Algorithms Lab
Code: PCC-CS494   **Semester: 4th**
Discipline: CSE   **Time: 2 Hours**
Date: February 21, 2023

1. Write a menu driven program in C or C++ to implement the following functions of a stack using array.

    a. ***void push(int)*** to add an item in the stack.

    b. ***int pop()*** to removes an item from the stack.

    c. ***int peek()*** to return top element of stack.

    d. ***int isFull()*** returns **1** if stack is full, else **0**.

    e. ***int isEmpty()*** returns **1** if stack is empty, else **0**.

    f. ***void print()*** to display the items in the stack.

2. Write a menu driven program in C or C++ to implement the following functions of a queue using array.

    a. ***void insert(int)*** to add an item in the queue.

    b. ***int del()*** to removes an item from the queue.

    d. ***int isFull()*** returns **1** if queue is full, else **0**.

    e. ***int isEmpty()*** returns **1** if queue is empty, else **0**.

    f. ***void print()*** to display the items in the queue.

3. Write a menu driven program in C or C++ to implement a queue using stack using array.

    A queue can be implemented using two stacks. Let queue to be implemented be ***Q*** and stacks used to implement be ***S1*** and ***S2***. Queue ***Q*** can be implemented in two ways: 1: by making delete operation costly and 2: by making insert operation costly.

    An implementation of a queue ***Q***, using two stacks ***S1*** and ***S2***, is given below (by making delete operation costly):

```
1 void insert(Q, x) {
2    push (S1, x);
3 }
```

```
1  void delete(Q){
2     if(isEmpty(S2)) then
3        if(isEmpty(S1)) then {
4           print("Queue is empty");
5           return;
6        }
7        else while (!(isEmpty(S1))){
8           x=pop(S1);
9           push(S2,x);
10       }
11    x=pop(S2);
12 }
```

Example:

Insert operation

Queue

Stack S1   Stack S2

Queue

Delete operation

Queue

Stack S1   Stack S2
Queue

Copy from
S1 to S2

Stack S1   Stack S2
Queue

Pop from S2

Copy from
S2 to S1

Stack S1   Stack S2
Queue

Stack S1   Stack S2
Queue