1111111111111111111111111111111111111111111111111111111111

National University of Computer & Emerging Sciences
FAST-Karachi Campus
CS218- Data Structures (Fall 2019)
Quiz#2

Dated: October 31, 2019 Marks: 25
Time: 20 min. Std-ID: _____Sol_____

Question No. 1

A FASTIAN suggested an array based implementation of two-stacks. Show how would you use an Array to keep two stacks S1 and S2 with the operations like: POP, PUSH, IsEmpty and IsFull. You are required to give the pseudocode of the four functions for both S1 and S2. [10]

```
#include<iostream>
using namespace std;
template <class T>
class DualStack{
        private:
         DynamicSafeArray<T>  *ds;
         int topLeft, topRight;
        public:
        DualStack()
                {
                ds= new DynamicSafeArray<T>(n);
                        topLeft=-1;
                        topRight= n;
                }

                ~DualStack()
                {
                        if (ds != 0)
                        delete ds;
                        ds=0;
                        topLeft=0;
                        topRight=0;
                }

        void PushLeftStack(T d)
        {
                if  (topLeft==-1) ||(topLeft < topRight -1)
                {       topLeft++;
                        ds[topLeft]=d;
                }

        }
```

```
void PushRightStack(T d)
        {  if (topRight==n) || (topRight > topleft+1)
                {
                        topRight--;
                        ds[topRight]=d;
                }
         }
        bool IsFull()
        {
                return (topLeft+1==topRight);
        }
        bool IsEmpty()
        {
                return ((topRight-topleft)>0);
        }
        T& PopRight()
        {       T *temp;
                if (!isEmpty())
                temp=ds[topRight];
                topRight++;
                return temp;
         }
         T& PopLeft()
        {
                T *temp;
                if (!isEmpty())
                temp=ds[topLeft];
                topLeft--;
                return temp;
        }

};
```

Question No. 2

Suppose you have a stack in which the values 1 through 5 must be pushed on the stack in that order, but that an item on the stack can be popped at any time. Give a sequence of push and pop operations such that the values are popped in the following order:

(a) 2, 4, 5, 3, 1

(b) 1, 5, 4, 2, 3

It might not be possible in each case. [10]

(a) Push (1), Pop, Push (2), Push (3), Push (4), Push (5), Pop, Pop? <Not Possible>

Alternate solution- Assuming all 1 through 5 elements are already pushed on stack, we need to pop elements in the following order if possible by using pop and retain elements and follow the same order for push again except the retain element.

(b) Push (1), Pop, Push (2), Push (3), Pop, Push (4), Push (5), Pop, Pop, Pop

Question No. 3

Convert the following infix expression into equivalent postfix expression. Using the algorithm discussed in class as an application of stack. [5]

(a-b*c-d)*a+b-(c/a)

The output will be: a b c * - d - a * b + c a / -