Question1.cpp

```cpp
// <----Lab 05 - Stacks---->

// Q1. Please write a program which performs the following tasks:
// a. Make a left to right scan of the postfix expression
// b. If the element is an operand push it on Stack
// c. If the element is operator, evaluate it using as operands the correct number from stack
// and pushing the result onto the stack
#include<iostream>
#include<string>

using namespace std;

class stack {
        float arr[20];
        int top=-1;
    public:
        stack() {
            for(int i=0; i<20; i++) {
                arr[i]=0;
            }
        }
        bool full() {
            return top==19;
        }
        bool empty() {
            return top==-1;
        }

        void push(float value) {
            if(!full()) {
                top++;
                arr[top]=value;
            } else {
                cout<<"Array is Full.\n";
                return;
            }
        }
        float pop() {
            if(!empty()) {
                int temp = arr[top];
                top--;
                return temp;
            } else {
                cout<<"Array is Empty.\n";
                return -1;
            }
        }

};

float solveeq(stack &obj,string exp) {
    string curr="";
    for(int i=0; i<exp.length(); i++) {
```

```
54          curr = exp[i];
55          if(curr>="0"&&curr<="9") {
56              obj.push(stoi(curr));
57          } else if(exp[i]=='+') {
58              float temp=obj.pop();
59              obj.push(obj.pop()+temp);
60          } else if(exp[i]=='-') {
61              float temp=obj.pop();
62              obj.push(obj.pop()-temp);
63          } else if(exp[i]=='/') {
64              float temp=obj.pop();
65              obj.push(obj.pop()/temp);
66          } else if(exp[i]=='*') {
67              float temp=obj.pop();
68              obj.push(obj.pop()*temp);
69          } else {
70              cout<<"Invalid Character!"<<endl;
71          }
72      }
73      return obj.pop();
74  }
75
76  int main() {
77      string exp;
78      stack e;
79      cout<<"Enter Postfix Expression without Exponents (Numbers will be read as Single Digits
    and as INTEGERS): \n";
80      cin>>exp;
81      cout<<"Answer of your Expression: "<<solveeq(e,exp)<<endl;
82  }
```