Fatima Jinnah Women University

Rabia Batool

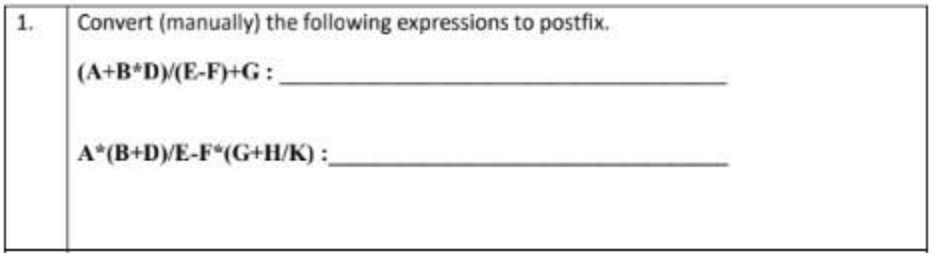
2022-BSE-064

Group#B

Data Structure & Algorithm

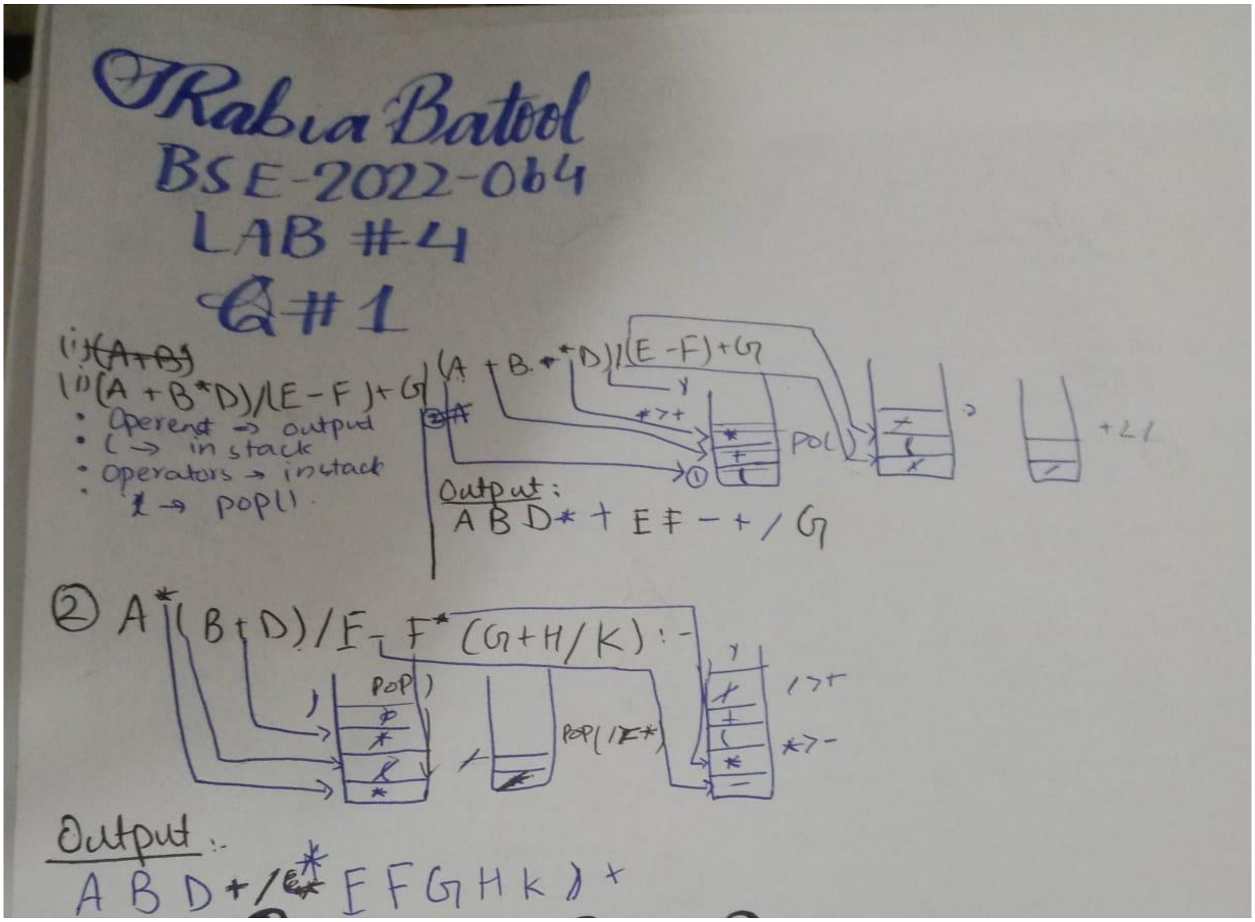
LAB#04

Submitted to Sir Rehan



1.	Convert (manually) the following expressions to postfix. **(A+B*D)/(E-F)+G : ____________________________________** **A*(B+D)/E-F*(G+H/K) :________________________________**

Answer:



Rabia Batool
BSE-2022-064
LAB #4
Q#1

(i) ~~(A+B)~~
(i) (A+B*D)/(E-F)+G

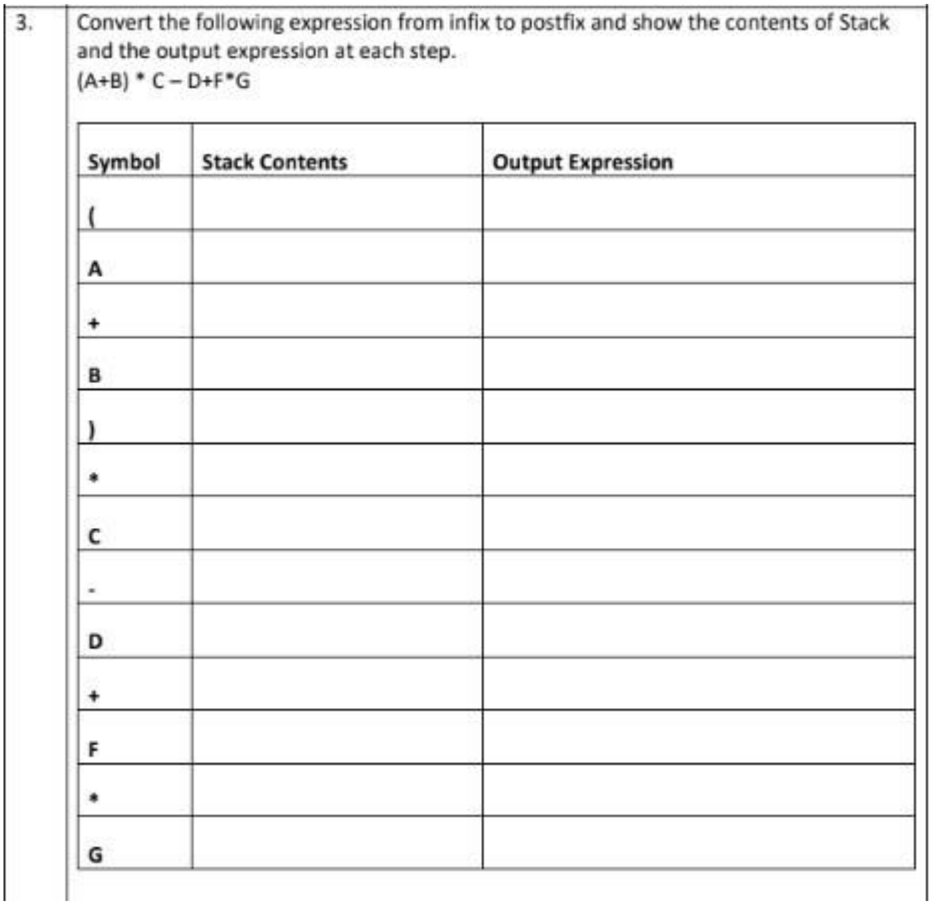
- Operent → output
- (→ in stack
- operators → in stack
-) → pop()

Output:
A B D * + E F - + / G

② A*(B+D)/E-F*(G+H/K) :-

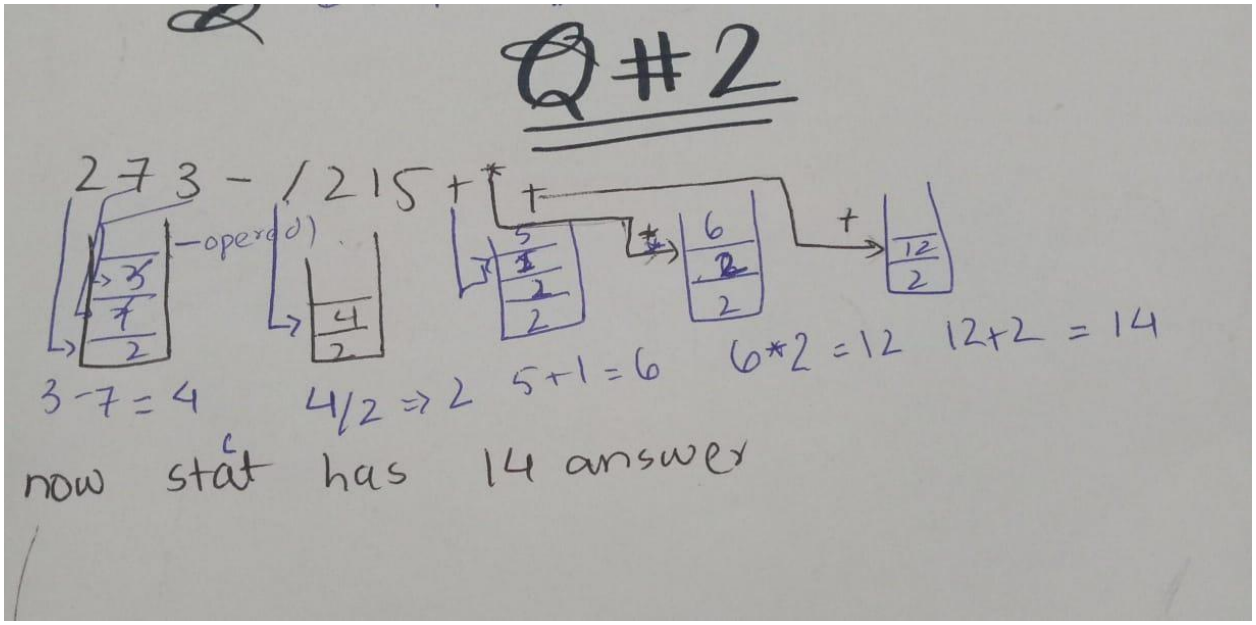
Output :-
A B D + / ~~E*~~ * E F G H K) +

(A + B * D)/(E-F)+G
①
*>+
POP( )
+2/
POP)
POP(/E*)
/>+
*>-



3.	Convert the following expression from infix to postfix and show the contents of Stack and the output expression at each step. (A+B) * C – D+F*G

Symbol	Stack Contents	Output Expression
(		
A		
+		
B		
)		
*		
C		
-		
D		
+		
F		
*		
G		

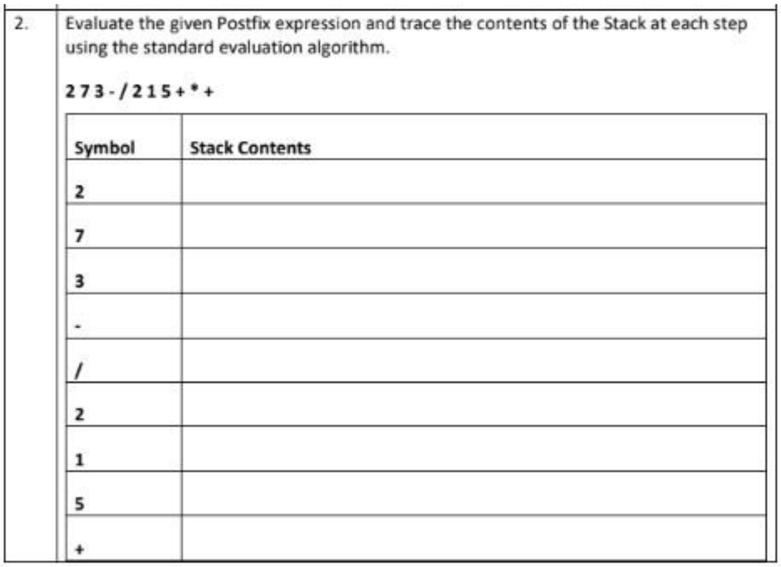


Q#2

2 7 3 - / 2 1 5 + * +

3-7 = 4 4/2 => 2 5+1 = 6 6*2 = 12 12+2 = 14

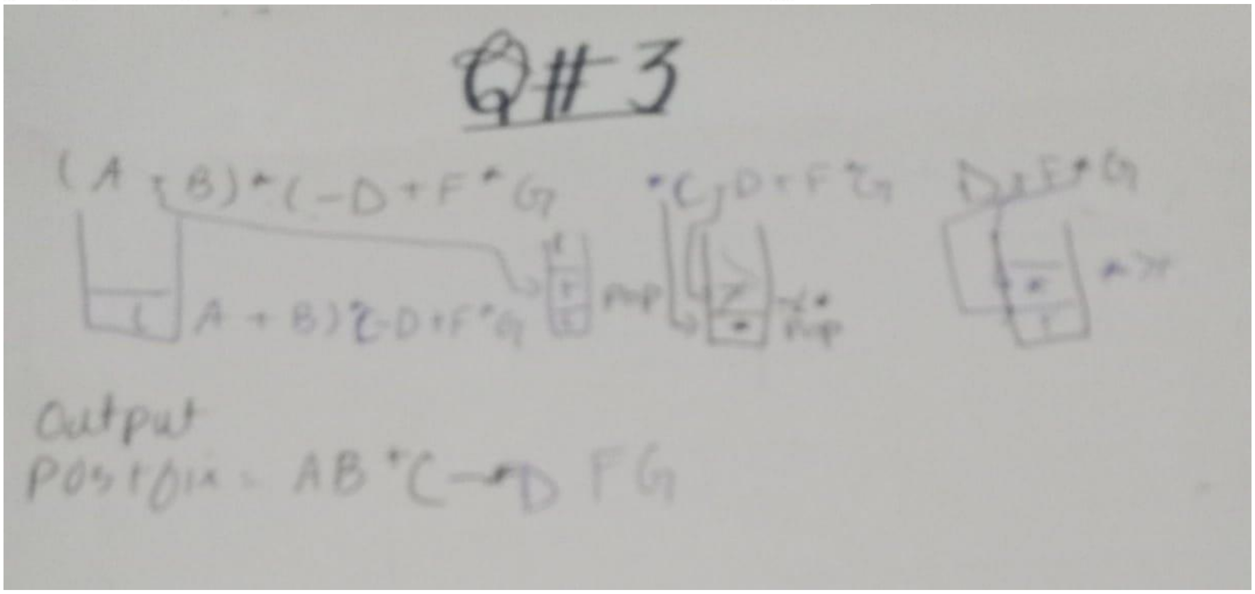
now stat has 14 answer

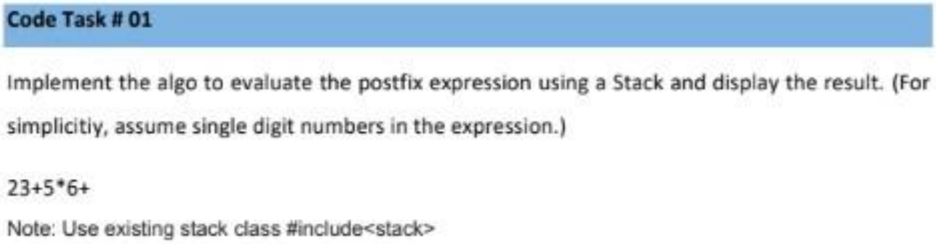


2. Evaluate the given Postfix expression and trace the contents of the Stack at each step using the standard evaluation algorithm.

2 7 3 - / 2 1 5 + * +

Symbol	Stack Contents
2	
7	
3	
-	
/	
2	
1	
5	
+	





Code Task # 01

Implement the algo to evaluate the postfix expression using a Stack and display the result. (For simplicitiy, assume single digit numbers in the expression.)

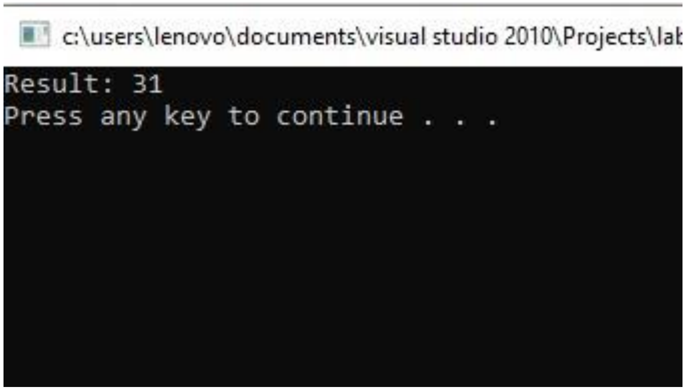
23+5*6+

Note: Use existing stack class #include<stack>

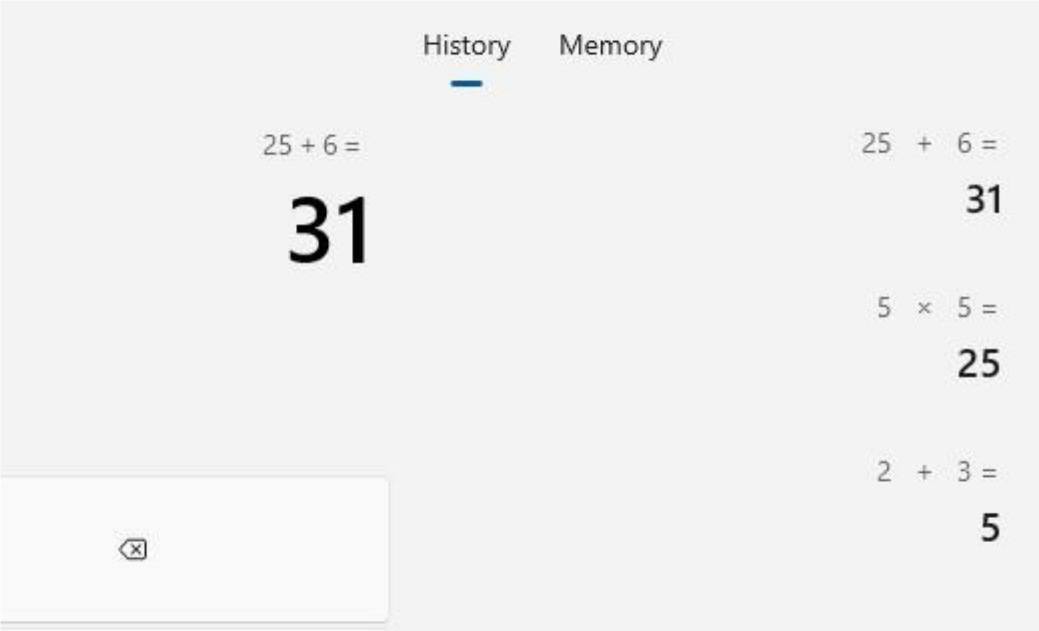
Code:

```
// lab 1.cpp : Defines the entry point for the console application.
//
```

```
#include "stdafx.h"
#include<iostream>
#include<string.h>
#include<stack>;
using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{stack <int>s;

       string str;
       str="23+5*6+";
       for(int i=0; i < str.length();i++){
              if((str.at(i)=='+')||(str.at(i)=='-')||(str.at(i)=='*')||(str.at(i)=='/')){
                     int a=s.top();
                     s.pop();
                     int b=s.top();
                     s.pop();
                     if(str.at(i)=='+'){
                            int c= a+b;
                            s.push(c);
                     }
                     if(str.at(i)=='-'){
                            int c= a-b;
                            s.push(c);
                     }
                     if(str.at(i)=='*'){
                            int c= a*b;
                            s.push(c);
                     }
                     if(str.at(i)=='/'){
                            int c= a/b;
                            s.push(c);
                     }

              }

              else{
                     int x=str.at(i)-'0';

                     s.push(x);

              }



}
       cout<<"Result: "<<s.top()<<endl;

       system("pause");
       return 0;
}
```



```
c:\users\lenovo\documents\visual studio 2010\Projects\lab
Result: 31
Press any key to continue . . .
```



25 + 6 =

31

25 + 6 =

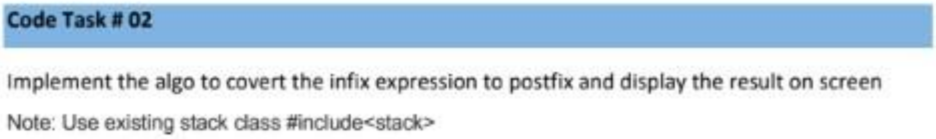
31

5 × 5 =

25

2 + 3 =

5



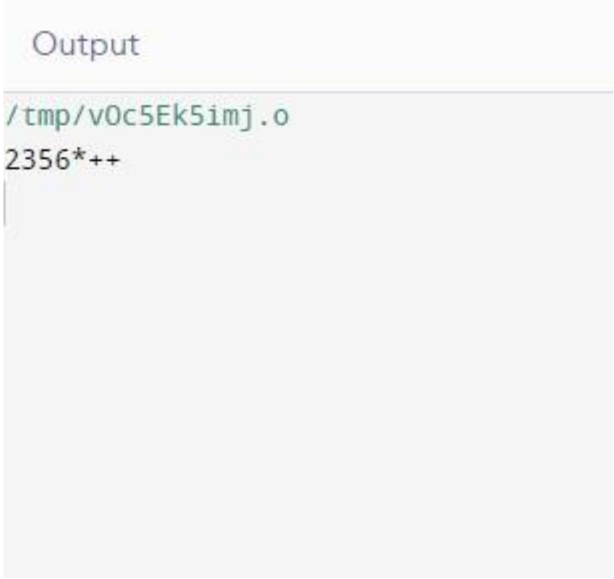
Code Task # 02

Implement the algo to covert the infix expression to postfix and display the result on screen

Note: Use existing stack class #include<stack>

```
#include <iostream>
#include <stack> using
namespace std;

int getPrecedence(char op) {
if (op == '*' || op == '/')
return 2;
    else if (op == '+' || op == '-')
return 1;     else         return 0;
}
 bool hasHigherOrEqualPrecedence(char op1, char op2)
{     return getPrecedence(op1) >
getPrecedence(op2);
```

```
}  int main() {
stack<char> s;

    string str = "(23+5*6+)";
    for (int i = 0; i < str.length(); i++) {
if (str.at(i) == '(') {
            s.push(str.at(i));
}
        else if (str.at(i) == ')') {
            while (!s.empty() && s.top() != '(') {
cout<<top();                 s.pop();
            }
            s.pop();
}
        else if (str.at(i) == '+' || str.at(i) == '-' || str.at(i) == '*' || str.at(i) ==
'/') {
            while (!s.empty() && s.top() != '(' && hasHigherOrEqualPrecedence(s.top(),
str.at(i))) {
                cout << s.top();
                s.pop();
            }
            s.push(str.at(i));
}         else {
            cout << str.at(i);
        }
    }      while
(!s.empty()) {
cout << s.top();
s.pop();
    }
    cout << endl;

    return 0;
}
```



Output

```
/tmp/vOc5Ek5imj.o
2356*++
```