

Practical No. 10 (Group D)

Name : Atharva B. Iparkar

Roll no : S211045

Class : S.E.

Div : A

Batch : A-2

Problem Statement :

Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions:

1. Operands and operator, both must be single character.
2. Input Postfix expression must be in a desired format.
3. Only '+', '-', '*' and '/' operators are expected.

Code :

```
#include <iostream>
```

```
#include <cstring> // Use for strlen
```

```
using namespace std;
```

```
class StackOp {
```

```
    char st[20], st1[20];
```

```
    int top;
```

public:

```
StackOp() {  
    top = -1;  
}  
void input();  
void push(char a);  
void pop();  
int pri(char b);  
};
```

```
int StackOp::pri(char b) {  
    if (b == '+' || b == '-') return 1;  
    if (b == '*' || b == '/') return 2;  
    return 0; // Default case for unsupported operators  
}
```

```
void StackOp::input() {  
    char ch[20];  
    top = -1;  
    int f = 1, l, i = 0, j = 0;  
    cout << "\nEnter the expression: ";  
    cin >> ch;  
    l = strlen(ch);  
    while (i < l) {
```

```

f = 1;

if (isalpha(ch[i]) || isdigit(ch[i])) {

    cout << ch[i]; // Outputting operand directly

    st1[j++] = ch[i];

}

if (ch[i] == '(') {

    push(ch[i]);

}

if (ch[i] == ')') {

    while (st[top] != '(') {

        cout << st[top];

        st1[j++] = st[top];

        pop();

    }

    pop(); // Remove '('

}

if (ch[i] == '+' || ch[i] == '-' || ch[i] == '*' || ch[i] == '/') {

    while (f == 1) {

        if (top == -1 || st[top] == '(') {

            push(ch[i]);

            f = 0;

        } else {

            if (pri(ch[i]) > pri(st[top])) {

                push(ch[i]);

            }

```

```

        f = 0;

    } else {

        cout << st[top];

        st1[j++] = st[top];

        pop();

    }

}

}

}

i++;

}

// Pop remaining operators in the stack

while (top != -1) {

    cout << st[top];

    st1[j++] = st[top];

    pop();

}

// Output the postfix expression

cout << "\nPostfix expression: ";

for (i = 0; i < j; i++) {

    cout << st1[i];

}

```

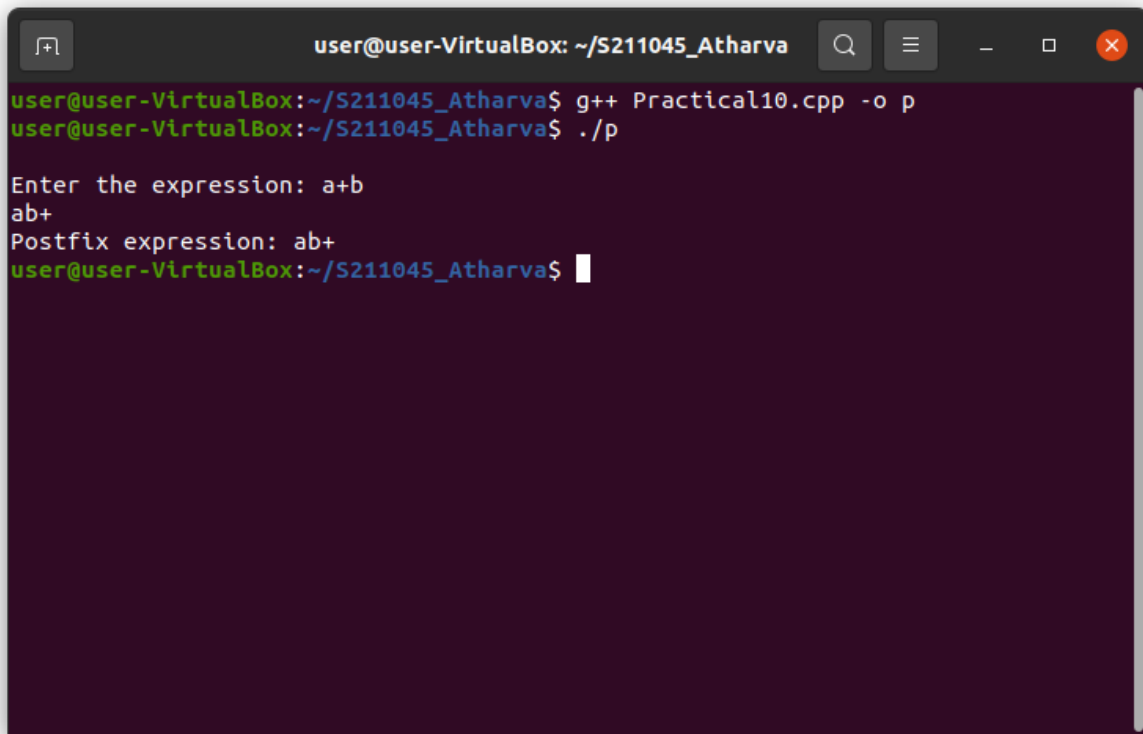
```
    cout << endl;
}
```

```
void StackOp::push(char a) {
    if (top >= 19) {
        cout << "Stack overflow\n";
        return;
    }
    st[++top] = a;
}
```

```
void StackOp::pop() {
    if (top < 0) {
        cout << "Stack underflow\n";
        return;
    }
    top--;
}
```

```
int main() {
    StackOp s;
    s.input();
    return 0;
}
```

Output :



```
user@user-VirtualBox: ~/S211045_Atharva
user@user-VirtualBox:~/S211045_Atharva$ g++ Practical10.cpp -o p
user@user-VirtualBox:~/S211045_Atharva$ ./p

Enter the expression: a+b
ab+
Postfix expression: ab+
user@user-VirtualBox:~/S211045_Atharva$
```

The image shows a terminal window with a dark background. The title bar at the top reads "user@user-VirtualBox: ~/S211045_Atharva". The terminal content shows the compilation of "Practical10.cpp" into an executable "p" using "g++". After running the program, it prompts for an expression. The user enters "a+b", and the program outputs "ab+", which is the postfix representation of the input expression. The prompt "Postfix expression: ab+" is displayed on the next line. The terminal ends with the user's shell prompt.