

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

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

```
#include<iostream>
#include<cstring>
#include<stack>
using namespace std;
int getWeight(char ch)
{
    switch (ch)
    {
        case '/':
        case '*': return 2;
        case '+':
        case '-': return 1;
        default : return 0;
    }
}
void infix2postfix(char infix[], char postfix[], int size)
{
    stack<char> s;
    int weight;
    int i = 0;
    int k = 0;
    char ch;
    while (i < size)
    {
        ch = infix[i];
        if (ch == '(')
        {
            s.push(ch);
            i++;
            continue;
        }
        if (ch == ')')
```

```

{
    while (!s.empty() && s.top() != '(')
    {
        postfix[k++] = s.top();
        s.pop();
    }
    if (!s.empty())
    {
        s.pop();
    }
    i++;
    continue;
}
weight = getWeight(ch);
if (weight == 0)
{
    postfix[k++] = ch;
}
else
{
    if (s.empty())
    {
        s.push(ch);
    }
    else
    {
        while (!s.empty() && s.top() != '(' && weight <=
getWeight(s.top()))
        {
            postfix[k++] = s.top();
            s.pop();
        }
        s.push(ch);
    }
    i++;
}
while (!s.empty())
{

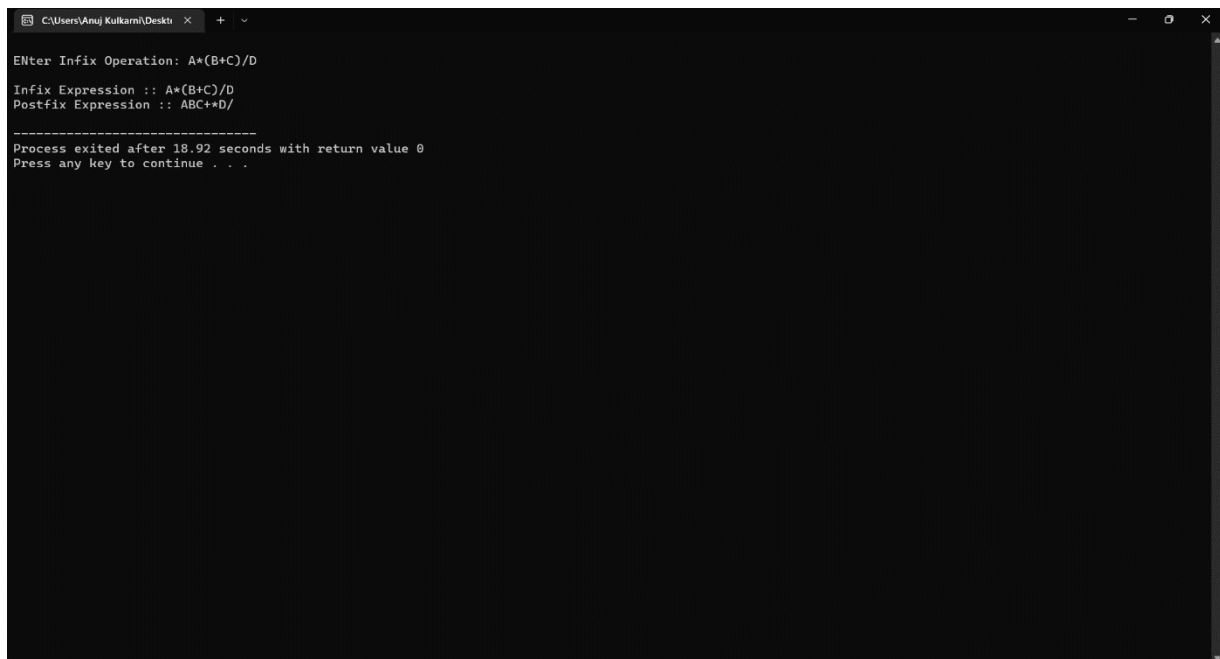
```

```

        postfix[k++] = s.top();
        s.pop();
    }
    postfix[k] = 0;
}
int main()
{
    char infix[100];
    cout<<"\nENter Infix Operation:";
    cin>>infix;
    int size = strlen(infix);
    char postfix[size];
    infix2postfix(infix,postfix,size);
    cout<<"\nInfix Expression :: "<<infix;
    cout<<"\nPostfix Expression :: "<<postfix;
    cout<<endl;
    return 0;
}

```

## Output :-



```

C:\Users\Anuj Kulkarni\Desktop>
Enter Infix Operation: A*(B+C)/D
Infix Expression :: A*(B+C)/D
Postfix Expression :: ABC+*D/
-----
Process exited after 18.92 seconds with return value 0
Press any key to continue . . .

```