PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

S.Y. B. TECH **Year:** 2024 – 25 **Semester:** I

Name: Sonawane Prachi Mahendra. **PRN:** 124B2B018

Department : Computer Engineering **Division:** B

Course : Data Structures Laboratory **Course Code:** BCE23PC02

Date: 5/10/24

Assignment – 8

- **Aim:**

  Write a program to convert infix expression to postfix, infix expression to prefix and evaluation of postfix and prefix expression.

- **Source Code:**

a)

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

int priority(char c)
{
    if((c=='*')||(c=='/'))
    {
        return 3;
    }
    if((c=='+')||(c=='-'))
    {
```

```
        return 2;
        }
        return 0;
}
string infixToPostfix(string s)
{
        string postfix="";
        int top=-1;
        char stack[10];
        for(char c:s)
        {
                if((c=='+')|| (c=='*')|| (c=='-')|| (c=='/'))
                {
                        if(priority(stack[top]) >= priority(c))
                        {
                                postfix+=stack
                           [top]; top--;
                        }
                   top++;
                  stack[top]=c;
                }

                else
                {
                postfix+=c;
                }
}
```

```
while(top!=-1)
{
        postfix+=stack[top]; top--;
}
cout<<"\npostfix expression:"<<postfix; return postfix;
 }

 string reverse1(string s)
 {
string rev="";
for(int i=s.size()-1;i>=0;i--)
{
        rev+=s[i];
}
          return rev;
}
 int main()
 {
    string s="a+b*c"; cout<<"Original
    String:"<<s; string rev=reverse1(s);
    cout<<"\nReverse of original:"<<rev; string
    i=infixToPostfix(rev);
   string f=reverse1(i);
cout<<"\nAgain reverse of postfix:"<<f;
 }
```

b) Infix to Postfix conversion:

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

int priority(char c)
{
        if((c=='*')||(c=='/'))
        {
                return 3;
        }
        if((c=='+')||(c=='-'))
        {
                return 2;
        }
        return 0;
}

void infixToPostfix(string s)
{
        string postfix="";
        int top=-1;
        char stack[10]; for(char c:s)
        {
```

```
            if((c=='+')|| (c=='*')|| (c=='-')|| (c=='/'))

            {

               if(priority(stack[top]) >= priority(c))

               {
                  postfix+=stack[top]; top--;
               }
                top++; stack[top]=c;
            }
            else

            {

               postfix+=c;
            }

}


while(top!=-1)

{


            postfix+=stack[top]; top--;
}

cout<<"postfix string:"<<postfix;

        }


        int main()
        {
           string s="a+(b*c)+d";
           infixToPostfix(s);
        }
```

- **Screen shots of Output:**

  1.

```
Output

/tmp/jaVhaCQOr1.o
Original String:a+b*c
Reverse of original:c*b+a
postfix expression:cb*a+
Again reverse of postfix:+a*bc

=== Code Execution Successful ===
```

  2.

```
Output

/tmp/xzDJCrnAic.o
postfix string:a(bc)*d++

=== Code Execution Successful ===
```

- **Conclusion:**

  Hence, we studied about how to convert infix expression into postfix expression and prefix expression.