
Assignment Name: Implementation of Infix to Postfix Expression
Class: MCA I Lab: CA Lab III (DS)

```
#include<iostream.h>
#include<conio.h>
#include<string.h>

class convert
{
      char infix[20],postfix[20],s[20];
      int i,p,top;
public:
      convert()
      {
            top=-1;
            i=p=0;
            cout<<"\nEnter infix Expression:";
            cin>>infix;
            strcat(infix,")");
            s[++top]='(';
      }
      int precedance(char);
      void post();
      void display();
};

int convert::precedance(char ch)
{
      switch(ch)
      {
            case '^':return 3;
            case '*':return 2;
            case '/':return 2;
            case '+':return 1;
            case '-':return 1;
            default: return 0;
      }
}

void convert::post()
{
      char ch;
      while(top!=-1)
      {
            ch=infix[i++];
            if((ch>='A'&&ch<='Z')||(ch>='a'&&ch<='z')||(ch>='1'&&ch<='9'))
             postfix[p++]=ch;
            else if(ch=='(')
             s[++top]=ch;
            else if(ch=='+'||ch=='-'||ch=='*'||ch=='/'||ch=='^')
            {
                  while(precedance(ch)<=precedance(s[top]))
                  postfix[p++]=s[top--];
                  s[++top]=ch;
            }
            else if(ch==')')
            {
                  while(s[top]!='(')
                  postfix[p++]=s[top--];
```

```
                top--;
            }
            else
            cout<<"\nWrong string";
      }
      postfix[p]='\0';
}

void convert::display()
{
      cout<<"\nPostfix Expression is :"<<postfix;
}

void main()
{
      clrscr();
      convert c;
      c.post();
      c.display();
      getch();
}

*/ Output */


Enter infix Expression:(a*b-(c+d/e^f)*h)

Postfix Expression is :ab*cdef^/+h*-

Enter infix Expression:a+2*5

Postfix Expression is :a25*+
```