**Assigment No:- 1.3.2**

**Title :- Implementation of program based on Infix to postfix**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#include<iostream.h>

#include<conio.h>

#include<string.h>

class CONVERT

{

char infix[],postfix[],s[];

int i,p,top;

public:

CONVERT()

{

top=-1;

i=p=0;

cout<<"\n Enter 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<<"\n Wrong string";

}

postfix[p]='\0';

}

void CONVERT::display()

{

cout<<"\n Postfix Expression is

void main()

{

clrscr();

CONVERT c;

c.post();

c.display();

getch();

}