**PROGRAM::**

#include<iostream>

#include<stack>

#include<string.h>

#include<math.h>

using namespace std;

char \*type1,\*type2,op1,op2,result;

int i,j,v;

char infix[50];

char postfix[50];

int size,top=-1;

class expr

{

stack<char>s;

public:

void infixtopostfix()

{

int weight;

int i=0,k=0;

char ch;

while(i<size)

{

ch=infix[i];

if(ch=='(')

{

s.push(ch);

//top++;

i++;

continue;

}

if(ch==')')

{

while(!s.empty() && s.top()!='(')

{

postfix[k++]=s.top();

s.pop();

// top--;

}

if(!s.empty())

{

s.pop();

//top--;

}

i++;

continue;

}

weight=getweight(ch);

if(weight==0)

postfix[k++]=ch;

else

{

if(s.empty())

{

s.push(ch);

//top++;

}

else

{

while(!s.empty() && s.top()!='(' && weight<=getweight(s.top()))

{

postfix[k++]=s.top();

s.pop();

//top--;

}

s.push(ch);

//top++;

}

}

i++;

}

while(!s.empty())

{

postfix[k++]=s.top();

s.pop();

//top--;

}

postfix[k]=0;

}

int getweight(char ch)

{

switch(ch)

{

case '^' : return 3; break;

case '/' :

case '\*' : return 2;

break;

case '+' :

case '-' : return 1;

break;

default : return 0;

}

}

void accept()

{

cout<<"\nEnter the expression to be converted into postfix\n";

cin.getline(infix,50);

size=strlen(infix);

/\*

cout<<"\n\nExpression is: \n";

for(i=0;i<size;i++)

{

cout<<infix[i];

}

\*/

}

void display()

{

cout<<"\n\nPostfix expression is\n";

for(i=0;i<size;i++)

{

cout<<postfix[i];

}

cout<<endl<<endl;

}

int evaluate(string s1)

{

char t,t1;

char val;

i=0;

t=s1[i];

while(t!='\0')

{

if(t>='0' && t<='9')

{

val=t-48;

//cout<<val;

s.push(val);

}

else

{

if(t=='+' || t=='-' || t=='\*' || t=='/' || t=='^')

op2=s.top(); s.pop();

op1=s.top(); s.pop();

switch(t)

{

case '+' : result=op1+op2;

break;

case '-' : result=op1-op2;

break;

case '\*' : result=op1\*op2;

break;

case '/' : result=op1/op2;

break;

case '^' : result=pow(op1,op2);

}

s.push(result);

}

i++;

t=s1[i];

}t1=result+48;

//cout<<"Result is";

//for(j=0;j<size;j++)

//cout<<postfix[i];

cout<<"Final result is : "<<t1;

}

};

int main()

{

expr ob1;

ob1.accept();

ob1.infixtopostfix();

ob1.display();

ob1.evaluate(postfix);

}

**OUTPUT:**

Enter the expression to be converted into postfix

(1+5)-(6\*11)

Postfix expression is

15+611\*-

Final result is : -60