**3. Write a C++ program that uses stack operations to convert a given infix expression into**

**its postfix equivalent, Implement the stack using an array.**

#include<iostream>

#include<conio.h>

#include<string.h>

using namespace std;

#define MAX 20

class conver

{

char stack[MAX];

int top;

public:

conver()

{

top=-1;

}

char pop();

void push(char item);

int prcd(char);

int isoperator(char);

void convertip(char[],char[]);

};

int conver::prcd(char symbol)

{

switch(symbol)

{

case '+':

case '-':return 2;

break;

case '\*':

case '/':return 4;

break;

case '^':

return 6;

break;

case '(':

case ')':

case '#':return 1;

}

}

int conver::isoperator(char symbol)

{

switch(symbol)

{

case '+':

case '-':

case '\*':

case '/':

case '^':

case '(':

case ')':return 1;

break;

default:return 0;

}

}

void conver::convertip(char infix[],char postfix[])

{

int i,symbol,j=0;

stack[++top]='#';

for(i=0;i<strlen(infix);i++)

{

symbol=infix[i];

if(isoperator(symbol)==0)

{

postfix[j]=symbol;

j++;

}

else

{

if(symbol=='(')

push(symbol);

else if(symbol==')')

{

while(stack[top]!='(')

{

postfix[j]=pop();

j++;

}

pop();//pop out (.

}

else

{

if(prcd(symbol)>prcd(stack[top]))

push(symbol);

else

{

while(prcd(symbol)<=prcd(stack[top]))

{

postfix[j]=pop();

j++;

}

push(symbol);

}//end of else.

}//end of else.

}//end of else.

}//end of for.

while(stack[top]!='#')

{

postfix[j]=pop();

j++;

}

postfix[j]='\0';//null terminate string.

}

int main()

{

conver ob;

char infix[20],postfix[20];

cout<<"Enter the valid infix string:\n";

cin>>infix;

ob.convertip(infix,postfix);

cout<<"The corresponding postfix string is:\n";

cout<<postfix;

return 0;

}

void conver::push(char item)

{

top++;

stack[top]=item;

}

char conver::pop()

{

char a;

a=stack[top];

top--;

return a;

}

**Output:-** 