C25: infix to post fix and evaluation of postfix expression

/\* Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions

i. Operands and operator, both must be single character.

ii. Input Postfix expression must be in a desired format.

iii. Only '+', '-', '\*' and '/ ' operators are expected. \*/

#include<iostream>

#include<strings.h>

#include<string.h>

#include<math.h>

using namespace std;

struct stack

{

char s[30];

int top;

}st;

int main()

{

char infix[30], exp[30];

int len, result, post(char exp[30]);

void intopost(char infix[30]);

cout<<"\nenter infix expression";

//cin.ignore();

//cin.sync();

cin.getline(infix,sizeof(infix));

cout<<"\n infix:"<<infix;

intopost(infix);

cout<<"\nenter postfix expression";

cin.getline(exp,sizeof(exp));

len=strlen(exp);

exp[len]='$';

result=post(exp);

cout<<"\nvalue of the expression is:"<<result;

return 0;

}

void intopost(char infix[30])

{

st.top=-1;

st.top=st.top+1;

st.s[st.top]='$';

char postfix[30],ch;

int i,j;

int instack(char), incomming(char);

void push(char item);

char pop();

j=0;

for(i=0;infix[i]!='\0';i++)

{

ch=infix[i];

while(instack(st.s[st.top])>incomming(ch))

{

postfix[j]=pop();

j++;

}

if(instack(st.s[st.top])!=incomming(ch))

push(ch);

else

pop();

}

while((ch=pop())!='$')

{

postfix[j]=ch;

j++;

}

postfix[j]='\0';

cout<<"\nthe postfix expression is =>"<<postfix;

}

int instack(char ch)

{

int priority;

switch(ch)

{

case '+':

case '-':

priority = 2;

break;

case '\*':

case '/':

priority = 4;

break;

case '^':

priority = 5;

break;

case '(':

priority = 0;

break;

case '$':

priority = -1;

break;

default:

priority=8;

}

return priority;

}

int incomming(char ch)

{

int priority;

switch(ch)

{

case '+':

case '-':

priority = 1;

break;

case '\*':

case '/':

priority = 3;

break;

case '^':

priority = 6;

break;

case '(':

priority = 9;

break;

case ')':

priority = 0;

break;

default:

priority=7;

}

return priority;

}

void push(char item)

{

st.top++;

st.s[st.top]=item;

}

char pop()

{

char e;

e=st.s[st.top];

st.top--;

return e;

}

int post(char exp[30])

{

char ch,\*type;

int result,val,op1,op2,flag=0;

int i=0;

st.top=0;

ch=exp[i];

while(ch!='$')

{

if(ch>='0' && ch<='9')

type="operand";

else if(ch=='+' || ch=='-' || ch=='\*' || ch=='/' || ch=='^')

type="operator";

if(strcmp(type,"operand")==0)

{

val=ch-48;

push(val);

}

else

if(strcmp(type,"operator")==0)

{

op2=pop();

op1=pop();

flag=1;

}

switch(ch)

{

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);

break;

}

if(flag==1)

{

push(result);

flag=0;

}

i++;

ch=exp[i];

}

result=pop();

return(result);

}

enter infix expressiona+b

infix:a+b

the postfix expression is =>ab+

enter postfix expression623+-382/+\*2^

value of the expression is:49