#include <stdio.h>

#include<ctype.h>

#include<string.h>

struct stack

{

char a[20];

int top;

}s;

void push(char c)

{

if(s.top==20)

printf("stack is full");

else

{

s.top++;

s.a[s.top]=c;

}

}

char pop()

{

char x;

if(s.top==-1)

{

return'0';

}

else

{

x=s.a[s.top];

s.top--;

return x;

}

}

char tops()

{

if(s.top==-1)

return'#';

else

return(s.a[s.top]);

}

int preced (char c)

{

if(c=='+'||c=='-')

return 1;

else if (c=='\*' || c=='/')

return 2;

else

return 3;

}

int islow(char c,char d)

{

if(preced(c)<=preced(d))

return 1;

else

return 0;

}

int isoperator(char c)

{

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

return 1;

else

return 0;

}

void main()

{

char infix[20],post[20],c;

int len,i,j;

s.top=-1;

printf("Enter the infix expression :");

gets(infix);

len=strlen(infix);

infix[len]=')';

infix[++len]='\0';

i=j=0;

push('(');

do

{

c=infix[i];

if (isalpha(c))

{

post[j]=c;

j++;

}

else if(c=='(')

{

push(c);

}

else if (isoperator(c))

{

while(isoperator(tops()) && islow(c,tops()))

{

post[j]=pop();

j++;

}

push(c);

}

else if(c==')')

{

while (tops()!='(')

{

post[j]=pop();

j++;

}

pop();

}

i++;

}

while(tops()!='#');

post[j]='\0';

printf("\npostfix from is %s",post);

getch();

}