#include<stdio.h>

#define size 100

char stack [size];

int top=-1;

void push(char item)

{

if(top>=size-1)

{

printf("OOPS!!! STACKOVERFLOW!!");

}

else

{

top++;

stack[top]=item;

}

}

char pop()

{

char item;

item=stack[top];

top--;

return(item);

}

int is\_operator(char symbol)

{

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

{

return 1;

}

else

{

return 0;

}

}

int presidence(char symbol)

{

if(symbol=='^')

{

return 3;

}

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

{

return 2;

}

else if(symbol=='+'||symbol=='-')

{

return 1;

}

else

{

return 0;

}

}

int main()

{

char infix[size],postfix[size],item,temp;

int i=0,j=0;

printf("ENTER THE INFIX ARITHMETIC EXPRESSION\n ");

gets(infix);

while(infix[i]!='\0')

{

item=infix[i];

if(item=='(')

{

push(item);

}

else if(item>='A'||item<='Z'||item>='a'||item<='z')

{

postfix[j]=item;

j++;

}

else if(is\_operator(item)==1)

{

temp=pop();

while(is\_operator(temp)==1&&presidence(temp)>=presidence(item))

{

postfix[j]=temp;

j++;

temp=pop();

}

push(temp);

push(item);

}

else if(item==')')

{

temp=pop();

while(temp!='(')

{

postfix[j]=temp;

j++;

temp=pop();

}

}

else

{

printf("INVALID ARITHMETIC EXPRESSION!!\n");

getch();

exit (0);

}

i++;

}

while(top>-1)

{

postfix[j]=pop();

j++;

}

postfix[j]='\0';

printf(" POSTFIX ARITHMETIC EXPRESSION : \n");

puts(postfix);

getch();

return 0;

}