#include<stdio.h>

#include<stdlib.h>

#include<string.h>

char postfix[100],infix[100],stack[100];

int isEmpty();

int isSpace(char);

int precedence(char);

void push(char);

char pop();

int top=-1;

void intopost();

void print()

{

printf("Postfix expression: ");

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

{

printf("%c",postfix[i]);

}

}

int main()

{

gets(infix);

intopost();

print();

return 0;

}

void intopost()

{

int i=0,j=0;

char next,symbol;

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

{

symbol = infix[i];

if(!(isSpace(symbol)))

{

switch(symbol)

{

case '(':

push(symbol);

break;

case ')':

while((next=pop())!='(')

{

postfix[j++] = next;

}

break;

case '+':

case '-':

case '\*':

case '^':

case '/':

while(!(isEmpty())&&(precedence(stack[top])>=precedence(symbol)))

{

postfix[j++] = pop();

}

push(symbol);

break;

default:

postfix[j++] = symbol;

break;

}

}

}

while(!(isEmpty()))

{

postfix[j++] = pop();

}

postfix[j] = '\0';

}

int isEmpty()

{

if(top == -1)

{

return 1;

}

else

{

return 0;

}

}

int isSpace(char symbol)

{

if(symbol == ' ' || symbol == '\t')

{

return 1;

}

else

{

return 0;

}

}

int precedence(char symbol)

{

switch(symbol)

{

case '^':

return 3;

break;

case '\*':

case '/':

return 2;

break;

case '+':

case '-':

return 1;

break;

default:

return 0;

break;

}

}

void push(char symbol)

{

top = top+1;

stack[top] = symbol;

}

char pop()

{

char ch;

ch = stack[top];

top = top-1;

return ch;

}