#include <iostream>

#include <string>

using namespace std;

int push(char s[], int &top, int elen, char input);

int pop(char s[], int &top, int elen);

int display(char s[], int top, int elen);

int priority(char op);

int opcomparison(char access, int &top, char stack[], int elen, string &postfix);

int main()

{

int top=-1;

string inp;

cout<<"Enter infix operation";

cin>>inp;

int elen=inp.size();

char stack[elen];

string postfix;

for(int i=0; i<elen; i++)

{

char access = inp[i];

opcomparison(access, top, stack, elen, postfix);

}

while(top!=-1)

{

postfix += stack[top];

pop(stack, top, elen);

}

cout<<"Output:"<<postfix;

}

int push(char s[], int &top, int elen, char input)

{

if (top + 1 < elen)

{

s[top+1] = input;

top++;

}

else

cout << "Stack is full" << endl;

return top;

}

int pop(char s[], int &top, int elen)

{

if (top != -1)

top--;

else

cout << "Stack is empty" << endl;

return top;

}

int display(char s[], int &top, int elen)

{

if(top !=-1)

{

for(int i=top; i>=0; i--)

cout<<s[i]<<" ";

cout<<endl;

}

else

cout<<"Stack is empty"<<endl;

}

int priority(char op)

{

switch (op)

{

case '^': return 2;

case '\*': case '/': return 1;

case '+': case '-': return 0;

default: return -1;

}

}

int opcomparison(char access, int &top, char stack[], int elen, string &postfix)

{

if (access == '(')

{

push(stack, top, elen, access);

}

else if (access == ')')

{

while (top != -1 && stack[top] != '(')

{

postfix += stack[top];

pop(stack, top, elen);

}

pop(stack, top, elen);

}

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

{

while (top != -1 && priority(stack[top]) >= priority(access))

{

postfix += stack[top];

pop(stack, top, elen);

}

push(stack, top, elen, access);

}

else

postfix += access;

}