/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates FATİH AYDIN @fatay

\* and open the template in the editor.

\*/

package systemsignature;

import java.util.Scanner;

class StackX{

private int maxSize; //Size of stack array

private char[] StackArray; //Stack array

private int top; //top of stack

public StackX(int s){ //CONSTURACTURE

maxSize = s; //set array size

StackArray = new char[maxSize]; //create array for stack

top = -1; //no items yet

}

public void push(char j){ //put item on top of stack

StackArray[++top] = j; //increment top, insert item

}

public char pop(){

return StackArray[top--];

}

public char peek(){ //peek at top of stack

return StackArray[top];

}

public boolean isEmpty(){ //is stack empty ?

return (top == -1);

}

public boolean isFull(){ //is stack full ?

return (top == maxSize -1);

}

}

public class SystemSignature {

// Global Variables --------------------------------------------------------

public static String only\_Operands = "";

public static String query = "";

public static String only\_One = "";

public static StringBuilder reset;

public static StringBuilder queryNew;

public static int signature[];

public static int pointOfWiev = 0;

public static void getParam(String str){

int t = 0;

boolean isZero = false;

System.out.println(str);

while(isZero == false && t < str.length()){

char c = str.charAt(t);

for (int k = 0; k < queryNew.length(); k++) {

char b = queryNew.charAt(k);

if (c == b) queryNew.setCharAt(k, '0');

}

String temp = queryNew.toString();

//System.out.println("TEMP = "+temp);

for (int i = 0; i < queryNew.length(); i++) {

char p = queryNew.charAt(i);

if(p!='0' && p!='+' && p!='\*') queryNew.setCharAt(i, '1');

}

char res = '\_';

StackX theStack = new StackX(queryNew.length());

for (int j = 0; j < queryNew.length(); j++) {

char ch = queryNew.charAt(j);

if(ch == '0' || ch == '1'){ theStack.push(ch);}

else{

switch(ch){

case '\*':

char b = theStack.pop();

char a = theStack.pop();

if(a =='1' && b == '1') theStack.push('1');

else theStack.push('0');

break;

case '+':

char r = theStack.pop();

char z = theStack.pop();

if( z == '1' || r == '1') theStack.push('1');

else theStack.push('0');

break;

}

}

}

res = theStack.pop();

//fatih aydin

if((char)res == '0'){

isZero = true;

signature[pointOfWiev]++;

queryNew = new StringBuilder(query);

pointOfWiev = 0;

//System.out.println("BEF ="+queryNew);

}else if((char)res == '1'){

isZero = false;

pointOfWiev++;

queryNew = new StringBuilder(temp);

//System.out.println("LAST = "+queryNew);

}

t++;

}

}

private static void permute(String str, int l, int r){

if (l == r){

getParam(str);

}else{

for (int i = l; i <= r; i++)

{

str = swap(str,l,i);

permute(str, l+1, r);

str = swap(str,l,i);

}

}

}

public static String swap(String a, int i, int j){

char temp;

char [] charArray = a.toCharArray();

temp = charArray[i];

charArray[i] = charArray[j];

charArray[j] = temp;

return String.valueOf(charArray);

}

static class InToPost {

private static StackX theStack;

private static String input;

private static String output = "";

public InToPost(String in) {

input = in;

int stackSize = input.length();

theStack = new StackX(stackSize);

}

public String doTrans() {

for (int j = 0; j < input.length(); j++) {

char ch = input.charAt(j);

switch (ch) {

case '+':

gotOper(ch, 1);

break;

case '\*':

gotOper(ch, 2);

break;

case '(':

theStack.push(ch);

break;

case ')':

gotParen(ch);

break;

default:

output = output + ch;

break;

}

}

while (!theStack.isEmpty()) {

output = output + theStack.pop();

}

return output;

}

public void gotOper(char opThis, int prec1) {

while (!theStack.isEmpty()) {

char opTop = theStack.pop();

if (opTop == '(') {

theStack.push(opTop);

break;

} else {

int prec2;

if (opTop == '+')

prec2 = 1;

else

prec2 = 2;

if (prec2 < prec1) {

theStack.push(opTop);

break;

}

else output = output + opTop;

}

}

theStack.push(opThis);

}

public void gotParen(char ch) {

while (!theStack.isEmpty()) {

char chx = theStack.pop();

if (chx == '(')

break;

else output = output + chx;

}

}

}

public static void main(String[] args) {

Scanner k = new Scanner(System.in);

String input = k.nextLine();

InToPost nevu = new InToPost(input);

query = nevu.doTrans();

System.out.println("Postfix is " + query + '\n');

only\_Operands += query.charAt(0);

for (int i = 1; i < query.length(); i++) {

char z = query.charAt(i);

boolean b = true;

if(z != '+' && z!= '\*'){

for (int j = 0; j < only\_Operands.length(); j++) {

char q = only\_Operands.charAt(j);

if(b==true && z==q)b = false;

}

if(b==true) only\_Operands += z;

}

}

signature = new int[only\_Operands.length()];

System.out.println("Your operands : "+only\_Operands + '\n');

queryNew = new StringBuilder(query);

reset = new StringBuilder(query);

int n = only\_Operands.length();

permute(only\_Operands, 0, n-1);

int all = 1;

for (int i = 2; i <= only\_Operands.length(); i++) {

all = all \* i;

}

System.out.print("YOUR SYSTEM SIGNATURE :");

for (int i = 0; i < signature.length; i++) {

System.out.print(signature[i] + "/" + all + " ");

}

}

}