package just\_random\_things;

import java.util.Scanner;

public class Calculator {

private String S1;

private String[] arr;

private int len;

private int[] arrPlus;

private int[] arrMin;

private int[] arrMul;

private int[] arrDiv;

private int i;

private int pl;

private int mi;

private int mul;

private int di;

private int s;

// Constructor

public Calculator(String S1, int len) {

this.S1 = S1;

this.i = 0;

this.pl = 0;

this.mi = 0;

this.mul = 0;

this.di = 0;

this.s=0;

this.len = len;

this.arr = new String[len];

this.arrPlus = new int[len];

this.arrMin = new int[len];

this.arrMul = new int[len];

this.arrDiv = new int[len];

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Note:- This calculator works only on unsigned numbers. ie No +8 or -7 as integers.\nThe operators allowed are + - \* /");

System.out.println("Enter Expression:");

String S1 = sc.nextLine();

S1=S1.trim();

int len = S1.length();

Calculator c1 = new Calculator(S1, len);

c1.eval();

sc.close();

}

void eval() {

char ch=S1.charAt(len-1);

int okay=0;

switch(ch) {

case '0':

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9': okay=1;

}

if(okay==0) {

System.out.println("Invalid Expression:");

return;

}

okay=0;

ch=S1.charAt(0);

switch(ch) {

case '0':

case '1':

case '2':

case '3':

case '4':

case '5':

case '6':

case '7':

case '8':

case '9': okay=1;

}

if(okay==0) {

System.out.println("Invalid Expression");

return;

}

for(i=1;i<len-2;i++) {

char chn=S1.charAt(i);

if(chn=='+'||chn=='-'||chn=='\*'||chn=='/') {

char chm=S1.charAt(i+1);

if(chm=='+'||chm=='-'||chm=='\*'||chm=='/') {

System.out.println("Invalid Expression");

return;

}

}

}

try {

// Converting String expression to char array

for (i = 0; i < len ;i++ ) {

int pos1 = S1.indexOf("+", i);

int pos2 = S1.indexOf("-", i);

int pos3 = S1.indexOf("\*", i);

int pos4 = S1.indexOf("/", i);

int minPos = len;

if (pos1 != -1 && pos1 < minPos) {

minPos = pos1;

}

if (pos2 != -1 && pos2 < minPos) {

minPos = pos2;

}

if (pos3 != -1 && pos3 < minPos) {

minPos = pos3;

}

if (pos4 != -1 && pos4 < minPos) {

minPos = pos4;

}

String num = S1.substring(i, minPos);

arr[s++]=num;

if(minPos==len) {

break;

}

switch (S1.charAt(minPos)) {

case '+':

arrPlus[pl++] = s;

arr[s++]="+";

break;

case '-':

arrMin[mi++] = s;

arr[s++]="-";

break;

case '\*':

arrMul[mul++] = s;

arr[s++]="\*";

break;

case '/':

arrDiv[di++] = s;

arr[s++]="/";

break;

}

i = minPos;

}

// Division

for (i = 0; i < di; i++) {

int cb = arrDiv[i] - 1;

int ca = arrDiv[i] + 1;

String chb = arr[cb];

String cha = arr[ca];

arr[arrDiv[i]] = String.valueOf(Double.parseDouble(chb) / Double.parseDouble(cha));

arr[cb] = null;

arr[ca] = null;

}

// Multiplication

for (i = 0; i < mul; i++) {

int cb = arrMul[i] - 1;

int ca = arrMul[i] + 1;

String chb = arr[cb];

String cha = arr[ca];

while (chb == null || chb.equals("")) {

chb = arr[--cb];

}

while (cha == null || cha.equals("")) {

cha = arr[++ca];

}

arr[arrMul[i]] = String.valueOf(Double.parseDouble(chb) \* Double.parseDouble(cha));

arr[cb] = null;

arr[ca] = null;

}

// Addition

for (i = 0; i < pl; i++) {

int cb = arrPlus[i] - 1;

int ca = arrPlus[i] + 1;

String chb = arr[cb];

String cha = arr[ca];

while (chb == null || chb.equals("")) {

chb = arr[--cb];

}

while (cha == null || cha.equals("")) {

cha = arr[++ca];

}

arr[arrPlus[i]] = String.valueOf(Double.parseDouble(chb) + Double.parseDouble(cha));

arr[cb] = null;

arr[ca] = null;

}

// Subtraction

for (i = 0; i < mi; i++) {

int cb = arrMin[i] - 1;

int ca = arrMin[i] + 1;

String chb = arr[cb];

String cha = arr[ca];

while (chb == null || chb.equals("")) {

chb = arr[--cb];

}

while (cha == null || cha.equals("")) {

cha = arr[++ca];

}

arr[arrMin[i]] = String.valueOf(Double.parseDouble(chb) - Double.parseDouble(cha));

arr[cb] = null;

arr[ca] = null;

}

for (i = 0; i < len; i++) {

if (arr[i] != null) {

System.out.print(arr[i]);

}

}

System.out.println();

}catch(Exception e) {

System.out.println("Something Invalid");

e.printStackTrace();

}

}

}