//Class: CSC 310

//Due Date: 10/26/18

//Author: Astro Reese Willoughby

//Asssignment: Project 1 - Postfix Expression Evaluation

package csc310project1;

import java.util.Scanner;

import java.util.Stack;

public class CSC310Project1 {

//evaluate a postfix expression and output the value

static int postfix(String exp)

{

//create a new, empty stack

Stack<Integer> stack = new Stack<>();

//This for loop will run through the expression and record the current

// character in char(c).

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

char c = exp.charAt(i);

//If c is a number, turn the character into an int type and

// push that int into the stack

if(Character.isDigit(c)){

int cd = c - '0';

stack.push(cd);

}

// If the scanned character is an operator, pop two

// elements from stack apply the operator

//If c is an operator (or other) then two values (val1 and

// val2) must be popped from the stack then have the operator applied

// to those two values

//Anything other than an operator (a parentheses or a space) will just

// be skipped over and the program will go onto the next character

// in the expression.

else{

int val1 = stack.pop();

int val2 = stack.pop();

if( c=='+'){

stack.push(val1+val2);

} else if (c=='-'){

stack.push(val2-val1);

} else if (c=='/'){

stack.push(val2/val1);

} else if (c=='\*'){

stack.push(val1\*val2);

} else {

continue;

}

}

}

//return the value that has been operated upon.

return stack.pop();

}

//main function

public static void main(String[] args){

Scanner in = new Scanner(System.in);

System.out.println("Input an expression using only numbers, spaces and "

+ "'+','-','\*','/' : " );

String expression = in.nextLine();

System.out.println("");

System.out.print("(" + expression + ") ");

System.out.println("will evaluate to: "+ postfix(expression));

}

}