import java.util.Scanner;  
  
public class Calculator {  
  
 public static void main(String[] args) throws Exception {  
  
 Scanner scanner = new Scanner(System.in);  
 System.out.println("Введите выражение в формате \"a + b\" или \"a - b\" или \"a \* b\" или \"a / b\"");  
 String expression = scanner.nextLine();  
  
 String[] tokens = expression.split(" ");  
 if (tokens.length != 3) {  
 throw new IllegalArgumentException("Выражение должно содержать два операнда и знак операции, разделенные пробелом.");  
 }  
  
 String a = tokens[0];  
 String b = tokens[2];  
 char operator = tokens[1].charAt(0);  
  
 // Определите тип цифр: арабские или римские  
 String digitsType = detectTokens(a, b);  
 if (digitsType == null) {  
 throw new Exception("Неподдерживаемая математическая операция!");  
 }  
  
 int operandA, operandB, result;  
 if (digitsType.equals("арабский")) {  
 operandA = convertToInt(a);  
 operandB = convertToInt(b);  
  
 result = calculate(operator, operandA, operandB);  
 System.out.println("Результат: " + result);  
 } else {  
 RomanNumeral roman = new RomanNumeral();  
 int num1 = roman.convertRomanToArabic(a);  
 int num2 = roman.convertRomanToArabic(b);  
  
 result = calculate(operator, num1, num2);  
 if (result < 0) {  
 // отрицательный результаты для римских цифр не допускается  
 throw new Exception("Результат не может быть меньше единицы для римских цифр!");  
 }  
  
 String romanResult = roman.convertArabicToRoman(result);  
 System.out.println("Результат: " + romanResult);  
 }  
  
 }  
  
 private static int calculate(char operator, int operandA, int operandB) {  
 int result = 0;  
 switch (operator) {  
 case '+':  
 result = operandA + operandB;  
 break;  
 case '-':  
 result = operandA - operandB;  
 break;  
 case '\*':  
 result = operandA \* operandB;  
 break;  
 case '/':  
 result = operandA / operandB;  
 break;  
 default:  
 throw new IllegalArgumentException("Неверный оператор: " + operator);  
 }  
 return result;  
 }  
 // Способ определения типа цифр: арабские или римские  
 private static String detectTokens(String a, String b) {  
 boolean isArabicA = a.matches("\\d+");  
 boolean isArabicB = b.matches("\\d+");  
  
 boolean isRomanA = RomanNumeral.isRomanNumeral(a);  
 boolean isRomanB = RomanNumeral.isRomanNumeral(b);  
  
 if ((isArabicA && isRomanB) || (isRomanA && isArabicB)) {  
 return null;  
 } else if (isArabicA && isArabicB) {  
 return "арабский";  
 } else {  
 return "римский";  
 }  
 }  
  
 // Способ преобразования строки в целое число  
 private static int convertToInt(String number) {  
 try {  
 int num = Integer.parseInt(number);  
 // Проверьте, находится ли число в допустимом диапазоне (1-10)  
 if (num < 1 || num > 10) {  
 throw new IllegalArgumentException("Операнды должны быть целыми числами от 1 до 10 включительно.");  
 }  
 return num;  
 } catch (NumberFormatException e) {  
 throw new IllegalArgumentException("Невозможно преобразовать строку в число: " + number);  
 }  
 }  
}  
  
class RomanNumeral {  
  
 private static final String[] ROMAN\_NUMERAL\_SYMBOLS = {"M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"};  
 private static final int[] ROMAN\_NUMERAL\_VALUES = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};  
  
 public static boolean isRomanNumeral(String str) {  
 // Проверьте,является ли строка допустимой римской цифрой  
 return str.matches("^(?=[MDCLXVI])M\*(C[MD]|D?C{0,3})(X[CL]|L?X{0,3})(I[XV]|V?I{0,3})" );  
 }  
  
public int convertRomanToArabic(String roman) {  
 // Преобразует римскую цифру в целое число  
 int result = 0;  
 int i = 0;  
 while (i < roman.length()) {  
 char currentSymbol = roman.charAt(i);  
 int currentValue = getValueOfRomanSymbol(currentSymbol);  
  
 if (i == roman.length() - 1) {  
 // Последний символ в строке, добавьте его значение к результату  
 result += currentValue;  
 break;  
 }  
  
 char nextSymbol = roman.charAt(i + 1);  
 int nextValue = getValueOfRomanSymbol(nextSymbol);  
  
 if (currentValue >= nextValue) {  
 // Текущий символ больше или равен следующему символу, добавьте его значение к результату  
 result += currentValue;  
 i++;  
 } else {  
 // Следующий символ больше текущего символа, вычтите значение текущего символа из результата  
 result += nextValue - currentValue;  
 i += 2;  
 }  
 }  
  
 return result;  
}  
  
public String convertArabicToRoman(int number) {  
 // Преобразуйте целое число в римскую цифру  
 StringBuilder result = new StringBuilder();  
 int i = 0;  
 while (number > 0) {  
 if (number >= ROMAN\_NUMERAL\_VALUES[i]) {  
 result.append(ROMAN\_NUMERAL\_SYMBOLS[i]);  
 number -= ROMAN\_NUMERAL\_VALUES[i];  
 } else {  
 i++;  
 }  
 }  
  
 return result.toString();  
}  
  
private int getValueOfRomanSymbol(char symbol) {  
 // Получить значение символа римской цифры  
 switch (symbol) {  
 case 'I':  
 return 1;  
 case 'V':  
 return 5;  
 case 'X':  
 return 10;  
 case 'L':  
 return 50;  
 case 'C':  
 return 100;  
 case 'D':  
 return 500;  
 case 'M':  
 return 1000;  
 default:  
 throw new IllegalArgumentException("Неверный символ Римской цифры: " + symbol);  
 }  
}  
}