2001/2002 ACM SOUTHERN CALIFORNIA REGIONAL SCHOLASTIC PROGRAMMING CONTEST

Problem 1
Interval Arithmetic

Interval arithmetic is a field of math wherein numbers, both constants and variables, are represented not as a single, exact value, but rather as a range or interval with lower and upper bounds. Under normal arithmetic, a scalar value represents a single point on the number line, such as 3.25. Under interval arithmetic, a scalar represents a segment of the number line, perhaps $[3, 5]$. Exact values, when represented as intervals, have identical upper and lower limits.

Arithmetical operations on two intervals involve the pairing of every point in one interval with every point in the other interval and performing the desired operation on each pair of points. The set of all points resulting from the operation form an interval. For example, adding $[3, 5]$ to $[-10, 1]$ yields $[-7, 6]$, the interval wherein all possible pairings of the addition of points from $[3, 5]$ and $[-10, 1]$ must lie. You are to write a program that can interpret single-line expressions of scalar interval constants with unary minus and the four basic operations of addition, subtraction, multiplication, and division. An example of each operation is shown below.

unary minus:	$-[-3, 5]=[-5, 3]$
addition:	$[3, 5] + [-10, 1]=[-7, 6]$
subtraction:	$[3, 5] - [-10, 1]=[2, 15]$
multiplication:	$[3, 5] * [-10, 1]=[-50, 5]$
division:	$[3, 5]/[-10, -0.1]=[-50, -0.3]$

Input to the program consists of one or more scalar intervals of the form [min,max] in infix expressions involving parentheses, unary minus (-), and binary addition (+), subtraction (-) , multiplication (*), and division (/). Parentheses may be nested. Spaces may occur within the line, but never between the interval brackets '[min,max]', nor after a unary minus. Your program need *not* handle exponential (E-format) numbers. No line will be longer than 80 characters. The standard rules of precedence (order of operations) apply; in decreasing precedence:

( )	parentheses
-	unary minus
* /	multiplication and division, left to right
+ -	addition and subration, left to right

For each line of input, produce a single line of output that is the resulting interval, in the form [min,max], where min $\leq$ max. The numbers in the output should be printed with three digits after the decimal point, and there should be no spaces within the interval. For expressions involving division by an interval that contains 0, output the string '`Division by zero`' rather than an interval.

Sample Input

```
  -[-3,5]
[3,5] + [-10,1]
[3,5]-[-10,1]
[3,5] * [-10,1]
( ([3,5] / [-10,-0.1]) / -[2,2] )
```

Sample Output

```
[-5.000,3.000]
[-7.000,6.000]
[2.000,15.000]
[-50.000,5.000]
[0.150,25.000]
```