CS3233 Test 1

Saturday, 12 August 2000, 11:45am-12:45pm — 100 marks

Submit your answers to the Local Judge.

106 Sum

50 marks

An evil primary school teacher, having taught the class on how to do arithmetic addition of two positive integers, gave every student a piece of paper with two very, very long numbers, and announced that the students were to submit the sum of the two numbers they are given as homework for the next day.

Unfortunately, since every homework sheet has different numbers, students cannot cheat. A student came to you and ask you for your help. You decided to scan in the numbers and write a program that sums them.

Input

The input has two lines, each containing a number. Both numbers are positive integers less than 10^{10^4}, given in decimal notation. None of the numbers start with leading zeroes.

Output

The output will be one line containing the sum. This line is terminated by a single newline (`\n`). There should be no leading zeroes.

Sample Input

```
1234567890987654321
9876543219123456789
```

Sample Output

```
11111111110111111110
```

107 Triangle

50 marks

Adapted from International Olympiad in Informatics 1994 day 1

7
3 8
8 1 0
2 **7** 4 4
4 **5** 2 6 5

The figure above shows a number triangle. Write a program that calculates the highest sum of numbers passed on a route that starts at the top and ends somewhere on the base.

Each step can go either diagonally down to the left or diagonally down to the right.

The number of rows in the triangle is between 1 and 100, inclusive.

The numbers in the triangle, all integers, are between 0 and 99, inclusive.

Input

The number of rows in the triangle is first read from the first line of the input, followed by the numbers in the triangle, as shown in the sample below.

Output

The highest sum is written as an integer in the output. This is followed by one route to get that sum, that is lexicographically smallest when written as a string of `L`s and `R`s (`L` for down-left and `R` for down-right). For example, the route `7-3-8-7-5` above (`LLRL` for down-left, down-left, down-right, then down-left) is more preferable than, say, `LRRL`, if the sums were equal.

Sample Input

```
5
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

Sample Output

```
30
7-3-8-7-5
```