University of Waterloo

Sun microsystems

Problem S1: Fix

A collection of words is *prefix-free* if no word is a prefix of any other word. A collection of words is *suffix-free* if no word is a suffix of any other word. A collection of words is *fix-free* if it is both prefix-free and suffix-free.

For this problem, a word is a sequence of lower-case letters of length between 1 and 25. A word X is a prefix of word Y if X consists of the first n characters of Y, in order, for some n. That is, the word "cat" has prefixes "c", "ca", and "cat". Similarly, a word X is a suffix of Y if X consists of the last n characters of Y, in order, for some n.

Your input will be $3N+1$ lines: the first line will be the number N, and the remaining $3N$ lines will be the N collections of 3 words each. (That is, lines 2, 3, and 4 compose the first collection, lines 5, 6, and 7 compose the second collection, and so on). Your output will be N lines, each line containing either Yes (if that collection of words is fix-free) or No (if that collection is not fix-free).

Sample Input

```
2
abba
aab
bab
a
ab
aa
```

Sample Output

```
Yes
No
```

Problem S2: TopYodeller

A yodel is a wordless song that shifts suddenly from a normal voice to a high, falsetto voice and back. Yodeling, present in many Swiss folk songs, is commonly heard throughout the Alps.

The TopYodeller contest invites the world's best yodellers to compete for a share of 100,000 Swiss Francs, with the best yodeller receiving 50,000 Francs. Each yodeller competes in a series of yodel rounds, and is assigned a score, by the judge, for their performance in the round. To keep the judge impartial, each yodeller is assigned a contestant number. Every yodeller competes in every yodel round. After all the yodel rounds are complete, the yodeller with the highest total score is declared the TopYodeller.

You have been hired by the TopYodeller Contest Co-ordinating Committee (CCC) to write a program which generates a scoreboard so that yodel fans around the world can track the progress of the best yodellers on the Internet.

After each round, each yodeller's cumulative score is calculated and a rank is assigned. A yodeller's rank is j+1 if j yodellers have a cumulative score higher than their score. Therefore, there may be multiple yodellers at the same rank.

Input

Each test case consists of one TopYodeller competition. The first line of the input contains two integers, n and k; n ($2 \le n \le 100$) represents the number of yodellers in the competition, and k ($1 \le k \le 100$) represents the number of yodel rounds in the competition.

Yodellers are assigned contestant numbers from 1 to n.

Next in the input are k lines, each line representing a yodel round.

Each line of input representing a yodel round contains n integers. These n integers give the scores assigned to the n yodellers by the judge. The first integer corresponds to the score given to yodeller number 1, and so forth. The score assigned to a yodeller in any given round is an integer between –1000 and 1000, inclusive.

Output

For the highest-ranked yodeller at the end of the competition, output "Yodeller x is the TopYodeller: score y, worst rank z" where x is the contestant number, y is their total score after the competition, and z is their worst rank at any time during the competition. If there is a tie for TopYodeller, the output should be one line per winning contestant, listed by increasing competitor number.

Sample Input

5 2
99 97 100 85 -4
95 97 100 62 1000

Sample Output

Yodeller 5 is the TopYodeller: score 996, worst rank 5

Explanation of Sample Input and Output

Does not have to be included with output.

After Round 1
Yodeller 3 has score 100, and is ranked 1.
Yodeller 1 has score 99, and is ranked 2.
Yodeller 2 has score 97, and is ranked 3.
Yodeller 4 has score 85, and is ranked 4.
Yodeller 5 has score -4, and is ranked 5.

After Round 2
Yodeller 5 has score 996, and is ranked 1.
Yodeller 3 has score 200, and is ranked 2.
Yodeller 1 has score 194, and is ranked 3.
Yodeller 2 has score 194, and is ranked 3.
Yodeller 4 has score 147, and is ranked 5.

University of
Waterloo

Sun microsystems

Problem S3: Spreadsheet

A spreadsheet consists of a number of "cells" set in a rectangular grid. Each is addressed with one letter from A to J (giving the row) and one number from 1 to 9 (giving the column). Thus the upper-left cell of the spreadsheet is A1, and the lower-right cell is J9.

Every cell has a value, and this can be specified in one of two ways:
1) as an integer from 0 to 1000
2) as a sum of the values of up to 10 other cells

Cell values may be interdependent (that is, A1's sum may depend on B6 which depends on C9), but a cell whose value depends on itself, directly or indirectly, is undefined (e.g., A1 depending on G8 depending on A1). Also, a cell whose value depends on an undefined cell is itself undefined. You are given the specification for all the cells in the spreadsheet. Compute and output the values of all of the cells.

Input
Input will consist of 10 lines, one per spreadsheet row. Each line contains 9 descriptions of a cell, which will be either an integer between 0 and 1000, or a sum of 1 to 10 distinct cell names separated by a "+" symbol (e.g. "A1+B5+D3").

Output
Output 10 lines, with 9 numbers per line, giving the value of every cell in the spreadsheet; if the cell is undefined, print an asterisk (*) in place of its value.
No cell's final value will exceed 1 000 000 000.

Sample Input
```
1 2 3 A1+A2+A3 A3+A4 A1+A4+A5 A8+A9 A9 A8
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

Sample Output
```
1 2 3 6 9 16 * * *
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
```

Problem S4: Space Turtle

Space Turtle is a fearless space adventurer. His spaceship, the *Tortoise*, is a little outdated, but still gets him where he needs to go.

The *Tortoise* can do only two things – move forward an integer number of light-years, and turn in one of four directions (relative to the current orientation): right, left, up and down. In fact, strangely enough, we can even think of the *Tortoise* as a ship which travels along a 3-dimensional co-ordinate grid, measured in light-years.

In today's adventure, Space Turtle is searching for the fabled Golden Shell, which lies on a deserted planet somewhere in uncharted space. Space Turtle plans to fly around randomly looking for the planet, hoping that his turtle instincts will lead him to the treasure.

You have the lonely job of being the keeper of the fabled Golden Shell. Being lonely, your only hobby is to observe and record how close various treasure seekers come to finding the deserted planet and its hidden treasure. Given your observations of Space Turtle's movements, determine the closest distance Space Turtle comes to reaching the Golden Shell.

Input

The first line consists of three integers *sx*, *sy*, and *sz*, which give the coordinates of Space Turtle's starting point. Space Turtle is originally oriented in the positive *x* direction, with the top of his spaceship pointing in the positive *z* direction, and with the positive *y* direction to his left. Each of these integers are between -100 and 100. The second line consists of three integers *tx*, *ty*, and *tz*, which give the coordinates of the deserted planet. Each of these integers are between -10000 and 10000. The rest of the lines describe Space Turtle's flight plan in his search for the Golden Shell. Each line consists of an integer, d, $0 \leq d \leq 100$, and a letter *c*, separated by a space. The integer indicates the distance in light-years that the *Tortoise* moves forward, and the letter indicates the direction the ship turns after having moved forward. 'L', 'R', 'U', and 'D' stand for left, right, up and down, respectively. There will be no more than 100 such lines.

On the last line of input, instead of one of the four direction letters, the letter 'E' is given instead, indicating the end of today's adventure.

Output

Output the closest distance that Space Turtle gets to the hidden planet, rounded to 2 decimal places. If Space Turtle's coordinates coincide with the planet's coordinates during his flight indicate that with a distance of 0.00. He safely lands on the planet and finds the Golden Shell.

Sample Input

0 0 0
1 1 1
2 L
2 L
2 U
2 U
2 L
2 L
2 U
2 E

Sample Output

1.41

University of Waterloo

Sun microsystems

Problem S5: Super Plumber

You are to write a program to play a video game in which Super Plumber (SP) navigates an obstacle course collecting prizes on the way to rescuing The Princess (TP).

The obstacle course is an m by n grid. SP starts at the bottom-left corner and makes his way to TP in the bottom-right corner. Some of the grid locations are occupied by obstacles through which SP cannot pass. Others are occupied by gold coins valued between $1.00 and $9.00.

The game is a traditional scroll game, which means that SP can move only to the right, up, or down. SP moves one grid location at a time, always to an adjacent location with no obstacle. He cannot occupy any location which he previously occupied - if he moves up, he cannot move down until he moves right; if he moves down he cannot move up until he moves right. SP collects the gold coins at locations he visits. You are to find the maximum possible total value of coins that SP can collect while rescuing TP.

Input has several test cases. The first line of each test case contains m and n, both integers not less than 2 or greater than 100. The grid is then given as m lines with n characters each. An obstacle is denoted by an asterisk ('*'); a coin is denoted by a digit ('1' through '9'); an empty location is denoted by a period ('.').
It is always possible for SP to rescue TP. A line containing 0 0 follows the last test case.

Output one line for each test case giving the amount of money that SP can collect. The sample input below contains two test cases. For the first case, SP can collect $27.00 with the following sequence of moves: *Up, Right, Down, Right, Right, Right, Right, Up, Right, Right, Down, Right, Right*. For the second case, SP can collect $34.00 with the following sequence: *Up, Right, Down*.

Sample Input

```
5 10
..3.......
..........
..7.**....
.9**...1..
..8..9....
2 2
99
88
0 0
```

Sample Output

```
27
34
```