2004 Canadian Computing Competition, Stage 2
Day 2, Question 1

Input file: `turtle.in` or standard input
Output file: `turtle.out` or standard output
Source file: `n:\turtle\turtle.___`

Return of Space Turtle

Remember Space Turtle, our fearless space adventurer? Last we encountered him, he was searching for the fabled Golden Shell in the Tortoise, his trusty spaceship.

Space Turtle has run out of fuel, but he thinks that he is very close to the Golden Shell. Unfortunately, due to a spatial anomaly (the sort you see on TV), both the Tortoise and the Golden Shell are trapped on a two-dimensional grid, travelling around and around in very strange orbits. These orbits always travel from lattice (integer co-ordinate) points on the grid to adjacent lattice points, and it takes exactly one minute to travel one unit of distance on the lattice. Both the Tortoise and the Golden Shell entered the anomaly at the same time, so you can really think of this as two things travelling around on a grid.

You can imagine that as the Tortoise and the Golden Shell are travelling in their respective orbits, the distance between them varies quite a bit. As the lonely keeper of the fabled Golden Shell, you have the task of observing the Tortoise once each minute (exactly when both you and the Tortoise are on lattice points) and recording how far away the Tortoise is. In particular, you want to determine the closest distance Space Turtle will ever be observed from the Golden Shell. (He might be closer when you're not looking, but that doesn't count.)

Input Description

The first line consists of two integers sx, and sy, which give the coordinates of Space Turtle's starting point in the anomaly, and a third integer, sm. Then, the orbit is described as a sequence of sm moves, each on a separate line. Each move consists of an integer, d, -100 = d = 100 and a letter c, separated by a space. The integer indicates the distance in light-years that the Tortoise moves, and the letter indicates the direction, either 'X' or 'Y', corresponding to the X and Y directions on the grid. There will be no more than 100 such lines. After this description, is an analogous description for the orbit of the Golden Shell: tx, and ty for the starting point, and tm on the first line, and then tm lines describing the orbit in the same manner as the first orbit. Both orbits are guaranteed to end at the starting location (so that they are cycles).

Output Description

Output the closest distance that you will ever observe between Space Turtle and the Golden Shell, to 2 decimal places. If you and Space Turtle collide on a lattice point at some

point, report 0.00.

Sample Input

```
0 0 4
-1 Y
-1 X
1 Y
1 X
1 0 4
-1 X
1 Y
1 X
-1 Y
```

Sample Output

```
1.00
```

2004 Canadian Computing Competition, Stage 2
Day 2, Question 2

Input file: `jenga.in` or standard input
Output file: `jenga.out` or standard output
Source file: `n:\jenga\jenga.___`

Jengaism

Jenga is a popular game involving a tower constructed of 1x1x3 blocks. Initially, this tower has 18 levels, each consisting of three blocks laid side to side. The blocks on alternating levels are oriented at right angles. Thus, each block touches all three of the blocks in the levels above and below it. Here is a picture of the actual game:

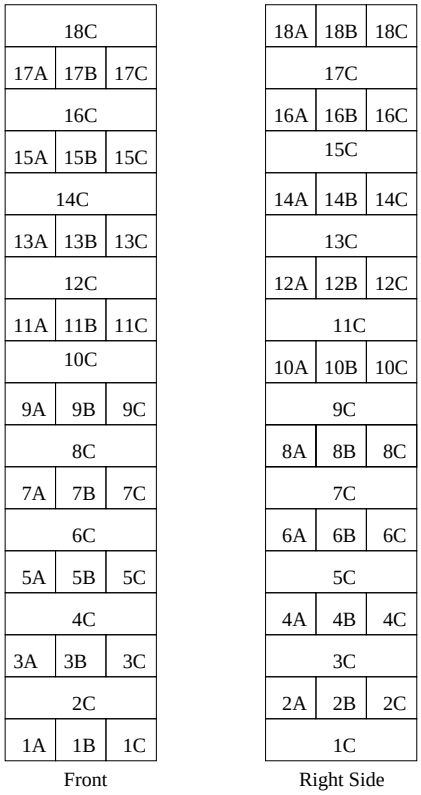
Play involves each player removing a block from somewhere in the tower, and putting it in a new position on top. The goal is to do this without knocking over the tower. Blocks are always removed from below the highest complete level, and the top level is always completed before a new level is started (at right angles, of course).

Write a program which reads sequential moves of a Jenga game and determines at what point the tower (or any part of it) falls or topples.

Note that a structure will topple if its center of gravity, projected onto its base, lies outside the convex hull of its support points. If the center of gravity lies exactly on the edge of this hull, we will assume that the structure is stable.

Input Specification

The first line contains N, the number of moves. The next N lines describe one move per line, with two locations separated by a single space. The first is the location of the block to be removed, and the second is where it will be put back. A location is specified as a number, specifying the level, and then a letter A-C, specifying its position in the level (left to right or front to back). For instance, the top level of the initial tower configuration consists of blocks at 18A, 18B, and 18C. Below is a diagram of the pieces, labelled with a front and right side perspective.



Output Specification

If the tower collapses after removing a block at location L, output "The tower collapses after removing L".

If the tower collapses after placing a block at location L, output "The tower collapses after placing L".

If all moves execute successfully (i.e., without causing the tower to fall), output "The tower never collapses".

Sample Input 1

```
4
```

```
6B 19B
7B 19A
17B 19C
17A 20B
```

Sample Output 1

```
The tower collapses after removing 17A
```

Sample Input 2

```
2
17C 19C
17A 19A
```

Sample Output 2

```
The tower never collapses
```

2004 Canadian Computing Competition, Stage 2
Day 2, Question 3

Input file: `orko.in` or standard input
Output file: `orko.out` or standard output
Source file: `n:\orko\orko.___`

Orko

Orko is a two-player card game. Each card has a colour (Red, Yellow, Green, or Black) and a value (1, 2, 3, 4, or 5). The deck contains twenty cards; one card having each distinct combination of colour and value.

Each player is dealt ten of the twenty cards. The game is played in ten rounds, and the objective is to win as many rounds as possible. In each round, one player, the player with 'the lead', plays one of his cards. The other player must play a card of the same colour, if he has one. If not, he may play any of his cards. The player with the lead wins the round if the other player has no card of the same colour, of if his card has a higher value. Otherwise the other player wins the round.

The objective of the game is to win as many rounds as possible. The lead for the first round is chosen arbitrarily; the lead for each subsequent round is given to the winner of the previous round.

Your job is to determine how many rounds each player will win, assuming that each player employs the strategy that maximizes his advantage.

The input contains several test cases. Each test case consists of one line of input, identifying the cards dealt to one of the players, "Player A." Each card is identified by a letter (`R`,`Y`,`G`,`B`) denoting its colour followed by a digit denoting its value (`1`,`2`,`3`,`4`,`5`). The other player, "Player B" receives the remaining cards in the deck. A line containing

* * * * * * * * * *

(10 stars, separated by spaces) follows the last test case.

For each test case, output a single line giving an integer between 0 and 10, the number of rounds won by Player A. Assume that Player A has the lead for the first round.

Sample Input

```
G1 G3 B2 R2 Y1 R3 R5 Y2 Y3 G5
* * * * * * * * * *
```

Output for Sample Input

```
3
```