Problem A : Linear Equation (blue balloon)

You have been asked to write a program that can solve a simple linear equation.

Standard Input

The first line of input contains a single integer **P**, ($1 \leq \mathbf{P} \leq 1000$), which is the number of data sets that follow. Each data set consists of a single line containing one simple linear equation. All equations are strings of less than 200 characters. Each equation will be in the form of **a*x*,** followed by a single space, followed by a sign "+", followed by **b**, followed by a single space ,followed by a sign "=", followed by a single space, followed by **c**.

$$ax + b = c$$

where ***x*** is the variable (real number) and **a**, **b**, **c** are positive integers.

Standard Output

For each data set, generate two lines of output. The first line will contain "Equation **n**" where **n** is the number of the data set. The second line will contain the following answer:

- ✓ If the equation has no solution, print "No solution.".
- ✓ If the equation has infinitely many solutions, print "More than one solution.".
- ✓ If the equation has exactly one solution, print "x = *solution*" where *solution* is replaced by the appropriate real number (printed to six decimals).

Print a blank line after each data set case.

Sample Input	Sample Output
5 2x + 3 = 4 124x + 20 = 160 123456x + 7 = 2000 0x + 2 = 3 0x + 2 = 2	Equation 1 x = 0.500000 Equation 2 x = 1.129032 Equation 3 x = 0.016143

	Equation 4 No solution. Equation 5 More than one solution.

Problem B: Vote

(red balloon)

Benin is organizing a presidential election the 06th march 2016. Several candidates submitted their applications for this presidential election. The CENA called "Commission Nationale Electorale Autonome" is responsible for managing the election.

According to Article 15 of the beninese electoral code, the CENA is responsible for the preparation, the organization, the supervision of voting and for the centralization of the results.

As a great programmer, you are asked to help the CENA to centralize the results of the presidential election.

Standard Input

The first line of input contains a single integer **P**, ($1 \leq \mathbf{P} \leq 1000$), which is the number of data sets that follow. Each data set consists of a line containing the number n of the candidates ($1 \leq \boldsymbol{n} \leq 100$), a space and the number m of results to centralize ($1 \leq \boldsymbol{m} \leq 1000$) and followed by n lines and m lines. The n lines contain the names of candidates, one per line. The m lines contain each a name **X** of one candidate, a space, the result **R** of the candidate and the center **C** of vote. **X** and **C** are strings that will contain at most 1000 characters. **R** is a positive integer.

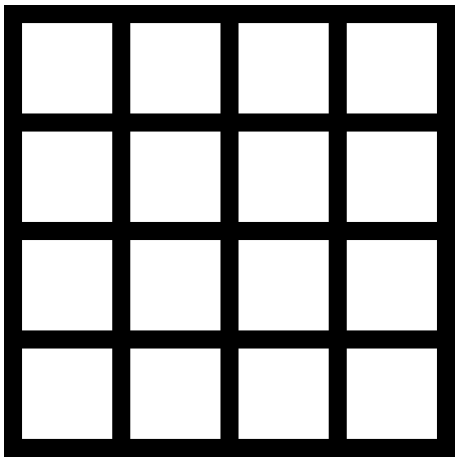
Standard Output

For each data set, if there is only one winner, generate one line of output with the text "VOTE i: THE WINNER IS " followed by the name of the winner, followed by space and followed by the total result of the winner. If not, generate the following output: "VOTE i: THERE IS A DILEMMA". i is the number of the data set.

Sample Input	Sample Output
2 3 4 Bignon Akwaba Sessi Bignon 1000 Gbegamey Sessi 1000 Yenawa Akwaba 5 Vodje Akwaba 996 Yenawa 2 3 Sena Sedjro Sedjro 6003 Malanville Sena 6000 Kpankpan Sena 3 Godomey	VOTE 1: THE WINNER IS Akwaba 1001 VOTE 2: THERE IS A DILEMMA

Problem C : Square
(yellow balloon)

Do you know a game called “La cave aux énigmes”? One of its questions is to find the number of squares contained in a grid square of length *l*. A grid square of length 4 will look like this:



The total number of squares that can be seen in this image is 30. Your task is to find the total number of squares which can be seen in an image of a grid square of length *l*.

Standard Input

The input will begin with a single integer **P** on the first line, indicating the number of cases that will follow.

The remaining lines of the input will consist of one integer ***l*** per line, which is the grid square length. All integers will be less than 1,000,000 and greater than 0.

You should process all integers and for each integer *l*, determine the total number of squares which can be seen in an image of a grid square of length *l*.

You can assume that no operation overflows a 32-bit integer.

Standard Output

For each integer *l*, you should output the total number of squares which can be seen in an image of a grid square of length *l*, with one line of output for each line of input.

Sample Input	Sample Output
4 1 2 3 4	1 5 14 30

Problem D : Prison Break (green balloon)

A prison has been built as a labyrinth.

The labyrinth is composed of huts labelled + or *. If you are in hut +, you can only move to another hut + near your hut. One hut is considered near one another if the two huts have a side in common. If you are in hut *, you can only move to another hut * near your hut.

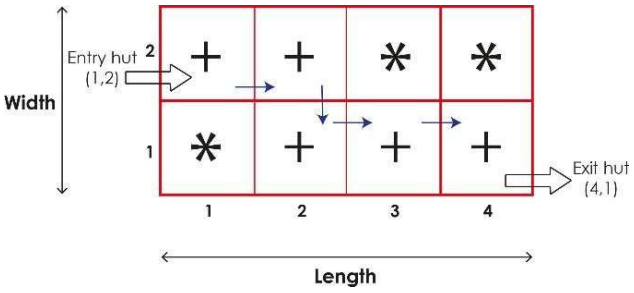
The labyrinth can be seen as a rectangle of huts of width **W** and length **L**. **W** and **L** are integers.

A hut is identified by its position on the horizontal side and by its position on the vertical side of the labyrinth.

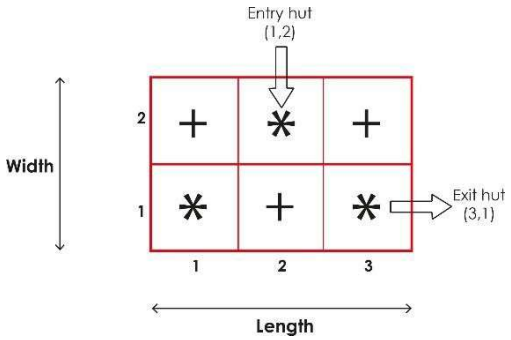
In this prison, all huts are “entry huts” but there is only one “exit hut”.

Given the labyrinth, the hut of the prisoner (called entry hut) and the exit hut, your task is to determine if the prisoner can escape.

In the example of Prison A (labyrinth of length 4 and width 2), the prisoner can escape. In the second example (labyrinth of length 3 and width 2), the prisoner cannot escape.



Prison A



Prison B

Standard Input

The input will begin with a single integer **P** on the first line, indicating the number of cases that will follow.

Each case begins with a single line made of 6 natural numbers with the following format:

L W A B C D where :

- **L** is the length of the labyrinth and **W** is the width of the labyrinth
- **A** is the length of the entry hut and **B** is the width of the entry hut of the prisoner
- **C** is the length of the exit hut and **D** is the width of the exit hut

followed by **W** lines containing **L** characters. Each character will be + or *.

Standard Output

For each prison, print YES if the prisoner can escape and NO if not.

Sample Input	Sample Output
2 4 2 1 2 4 1 ++** *+++ 3 2 2 2 3 1 +*+ *+*	YES NO

Problem E : X X glued
(white balloon)

Do you know the game "X X glued" practiced on the registration plates of Benin cars by children?

A beninese car registration number is written in the form of VY ABCD RB or Y ABCD RB. For example, AC 2554 RB and X 6006 RB are beninese car registration numbers. The game "X X glued" is to quickly verify that there is in the number two identical consecutive digits or not. The winner is the first child that pronounces "X X collés" and shows the car with his finger.

Standard Input

The input contains several lines and ended with the # character on a single line. Other lines each contain a single registration number as VY ABCD RB or Y ABCD RB.

Standard Output

For each registration number, your program should display on one line, all the "X X glued" in the format shown in the example. If there is no "X X glued" in the number, the program should display nothing.

Sample Input	Sample Output
X 6006 RB AC 2233 RB T 2000 RB F 2345 RB AB 4444 RB #	0 0 glued 2 2 glued and 3 3 glued 0 0 glued 4 4 glued