CCSS Programming Contest 2014

Exclusively for students attending the
Challenges in Computer Science Seminar
2013–2014 at Leiden University

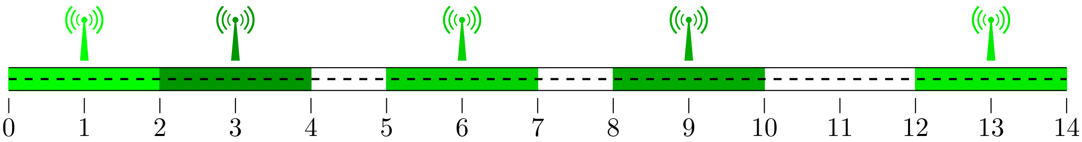
Problems

This is not a blank page

A Antennas

The Automobile Association (AA) is working on the AAAAA project: the Automobile Association's Antennas Along A272 project. This project entails placing antennas along the A272 so as to provide drivers with live traffic information. Some antennas have already been placed in previous projects, but additional antennas might be needed for this project.

The entire highway is L metres long. Each antenna provides coverage within a range of D metres around that antenna. Since the A272 is basically just a straight line, we will model it as the x-axis. Thus, an antenna located at the point x will cover the closed interval $[x - D, x + D]$, including its endpoints $x - D$ and $x + D$. For instance, with $D = 1$ and $L = 14$ we might have the following configuration:



The AA has provided you with the coordinates of the antennas in place, as well as the range D of a single antenna and the total length of the highway. Your task is to calculate the minimum number of additional antennas required to cover the entire A272. Note that antennas do not have to be placed at an integer number of metres from the beginning of the highway. However, the existing antennas are all located at integer locations. For instance, in the example above, three more antennas are needed. A possible solution is to put one antenna at $x = 4.5$, one at $x = 7.5$ and one at $x = 11$. (This is the third test case in the samples below.)

Input

The input starts with a line containing an integer T, the number of test cases. Then for each test case:

- One line containing three integers N, D and L: the number of antennas already in place, the range of a single antenna, and the length of the highway. In every test case one has $0 \leq N \leq 10^5$, $1 \leq D \leq 10^6$ and $1 \leq L \leq 10^9$.
- One line with N integers $a_1, \ldots, a_N$: the locations of the existing antennas. These coordinates satisfy $0 \leq a_1 < a_2 < \ldots < a_{N-1} < a_N \leq L$. The highway runs from $x = 0$ through $x = L$.

Output

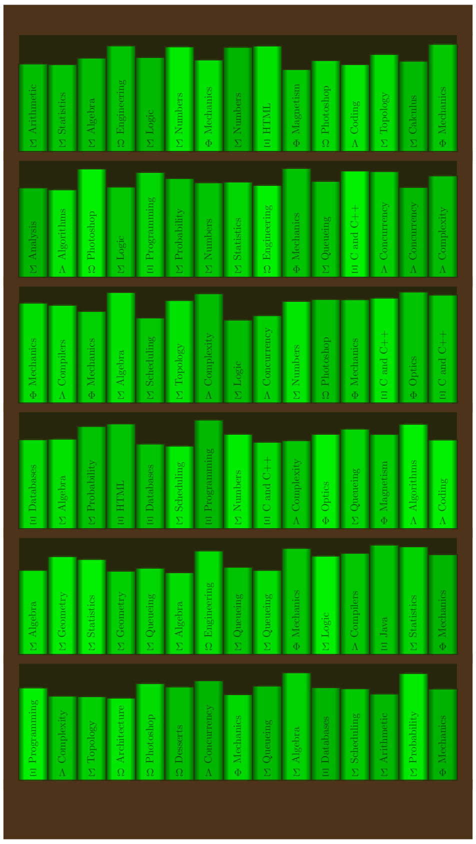
For each test case, output one line with a single integer M, the minimum number of additional antennas required for full coverage.

Sample input and output

Input	Output
3 1 5 20 10 3 4 24 4 12 20 5 1 14 1 3 6 9 13	2 0 3

This is not a blank page

B Books



Bas has just completed the gruelling task of sorting the books from his shelves by title. However he just realised that this is not a very useful ordering, as he often looks by subject instead. He therefore has to reorder his books once again.

Bas currently owns N books. The books are kept on one or more shelves, each of which can hold M books. He is going to sort the books by subject. If multiple books have the same subject, they are sorted by title. Bas then puts the first M books on the first shelf, the next M books on the second shelve, and so on. Your task is to determine which books go on which shelves.

The subject and the title of every book will consist of a non-empty string of uppercase letters (`A`–`Z`), lowercase letters (`a`–`z`) and underscores (`_`). Sorting is done as in most dictionaries: underscores are ignored, and no distinction is made between lower and uppercase letters. For instance, we consider "`Abundance`" and "`A Bun Dance`" to be the same title. Moreover, short words go before long words, in the sense that "`Themes`" goes before "`The Message`". We will make sure that no two books in the input go in the exact same place: the subject or the title will be different (not just by underscores or uppercase letters, but actually different).

Input

The input starts with an integer T, the number of test cases. Then, for each test case:

- A line containing two integers N and M satisfying $1 \leq N \leq 1000$ and $1 \leq M \leq 1000$.
- N lines describing the books currently in the bookcase. Each line describes subject and title in that order, separated by a space, a colon (`:`) and another space. The total number of characters per line is at most 100 (not counting the newline character after the end of the line). No subject or title begins or ends with an underscore, and there will never be two or more consecutive underscores. Every title and every subject will contain at least one letter.

In every test case, the books are given in order of title.

Output

For each test case, output the following data:

- One line containing an integer K, the number of shelves needed.
- K lines describing the subjects of the first and the last book on each shelf. Follow the output format described below.

Sample input and output

Input

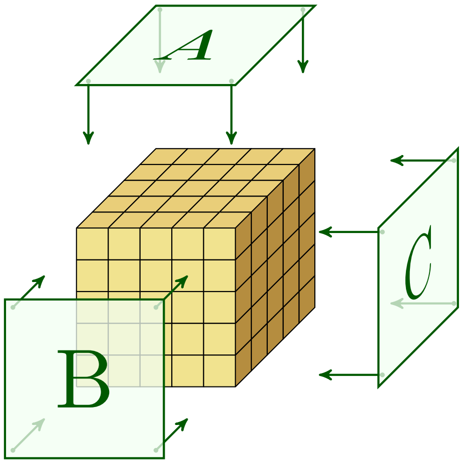
```
3
4 2
Recreatieve_wiskunde : Getallen_zijn_je_beste_vrienden
Informatica : Introduction_to_Algorithms
Logica : Logicomix
Speltheorie : Winning_Ways_for_your_Mathematical_Plays
7 3
Childrens_books : A_Bun_Dance
Economy : Abundance
Literature : Flatland
Children : How_to_raise_your_kids
Economy : The_Future_of_Bitcoin
Economy : Themes
Economy : The_Message
6 2
Zz : A
z_z : b
Zz : c
z_Z : d
zz : E
Zz : f
```

Output

```
2
Shelf 1: Informatica - Logica
Shelf 2: Recreatieve_wiskunde - Speltheorie
3
Shelf 1: Children - Economy
Shelf 2: Economy - Economy
Shelf 3: Literature - Literature
3
Shelf 1: Zz - z_z
Shelf 2: Zz - z_Z
Shelf 3: zz - Zz
```

In the third test case, the subjects of the books are considered to be the same for the sake of sorting the books. Thus, the order of the books is determined by their titles. Even though the subjects are considered to be the same, you should apply the proper spacing and capitalisation to the output. (Any answer other than the one presented above is considered to be wrong.)

C Counting Cardboard Cubes



The police have just arrived at a storage unit that is currently owned by a notorious bank robber called Carlos Costanza. They find a large collection of cardboard boxes, all of which have the same size (approximately $12.5 \times 12.5 \times 12.5$ inch), that are stacked together in neat piles. The police want to know how many boxes there are in total, but there are way too many boxes to count by hand. The chief of police therefore decides to take pictures of the top, front and right side views of the stack, as indicated in the figure on the right.

The 125 cubes in the figure are merely illustrative: in reality the situation is a little bit different. The letters A, B and C represent the places from which pictures are taken, and the pictures have not been rotated. This means that the bottom of the letter A corresponds with the bottom of the top view picture, and that the cubes located at the bottom of view A correspond with some of the cubes visible from the front side view B.

The chief of police now realises that the pictures might not contain enough information to count the exact number of boxes, since there might be different configurations that lead to the same three pictures. However, the chief of police thinks that he might still be able to determine the maximum number of boxes currently stacked in Carlos Costanza's storage unit. He doesn't know how to do that, so he calls for your help. You are given the three pictures taken by the police, and your task is to determine the maximum number of boxes that can be present.

Input

The input starts with a single integer T: the number of test cases. Then for each test case:

- One line with an integer D satisfying $1 \leq D \leq 100$, denoting the number of boxes in every direction. Thus, all the boxes fit within a $D \times D \times D$ cube.
- D lines showing the three pictures side by side. The leftmost picture is view A, the middle picture is view B and the rightmost picture is view C. In every picture, a hash (`#`) is used to represent a box and a period (`.`) is used to denote an empty space. In each line, the views are separated by a single space between two pictures, but no additional whitespace at the end of the line. That is: every line contains D characters, a space, D more characters, another space and finally yet another D characters.

The pictures are guaranteed to be accurate. In other words: some combinations of views A, B and C are invalid, but such combinations will not occur as a test case. (In particular, views B and C obey the laws of gravity.) Thus, there will always be at least one solution.

Output

For every test case, output the maximum number of boxes that can be present.

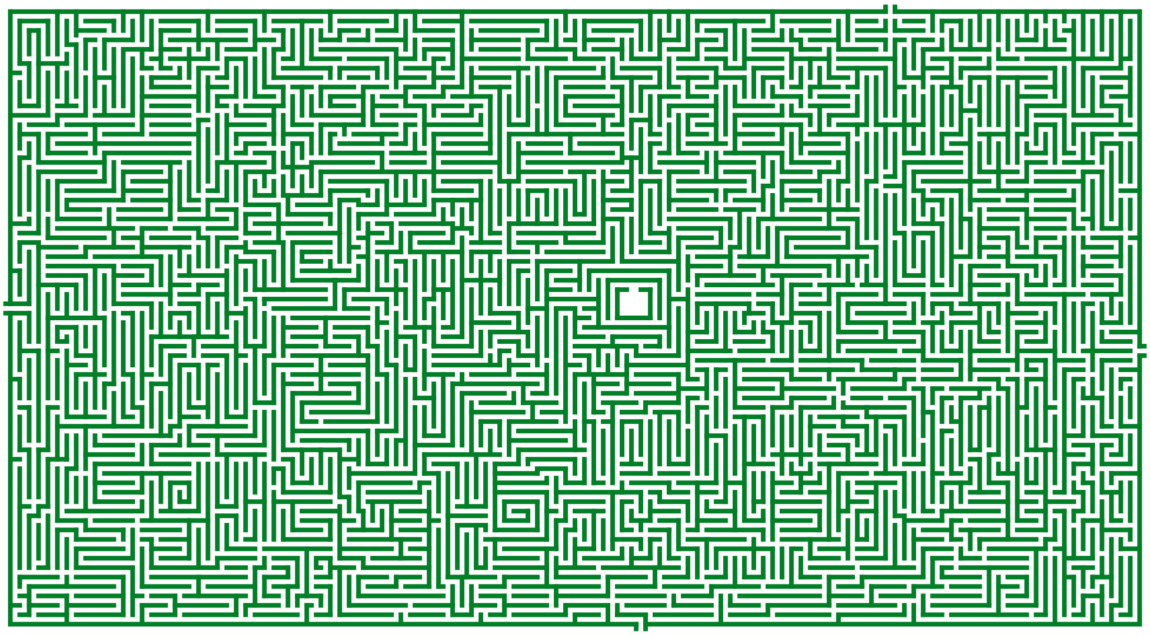
Sample input and output

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>4
5
....#.. .#...
.#..# ..#.. .#...
....#.. .#...
..#.. ..#.# .#.#.
..... .##.# .#.#.
5
#####
#####
##.## ##### #####
#####
#####
4
....
....
....
....
6
.....# .....# .....#
....#. ....## ....##
...#.. ...### ...###
..#... ..#### ..####
.#.... .##### .#####
#..... ###### ######</pre></td><td><pre>8
120
0
21</pre></td></tr>
</table>

In the second sample, the views do not provide enough information in order to uniquely determine the number of boxes (but there are 120 boxes at most). In the other samples, there is only one configuration of boxes leading to the given pictures.

D Distance

When you woke up this morning, you found yourself stuck in a maze. You have been walking around for hours but still haven't found the exit. Suddenly, you stumble upon a map, similar to the one depicted below. Use it to find the shortest way out.



Input

The input starts with a line with a single integer T, the number of test cases. Then, for each test case:

- One line containing two integers R and C, satisfying $3 \leq R, C \leq 500$. These denote the number of rows and columns, respectively.
- After that, R more lines follow containing C characters each, describing the maze. There are only three possible characters, each of which is one of the following:
  - A hash (`#`) denoting a wall.
  - A period (`.`) denoting an empty space.
  - A capital `Y`, the place where you are standing.

There will be exactly one letter `Y` per test case, and you are always standing on an empty space. You can move in four directions, and you obviously cannot move through walls.

Some of the outer cells are walls, but there might be several open spaces along the edge where you can leave the maze. There will always be at least one path leading out of the maze.

Output

For each test case, output a shortest path from your current location to an empty square on the edge of the maze. If there are multiple shortest paths, output any of them. Format the output like the input, replacing every period along your shortest path with the capital letter `O`. Print an extra blank line after each test case (including the last one).

Sample input and output

Input	Output
2	........####O.......
8 20	............O..#....
........####........	....#.......O.#.....
...............#....	#.........YOO......#
....#.........#.....	....
#.........Y........#	...#.............#..
....	...#................
...#.............#..	........####........
...#................	
........####........	#########
7 9	OOO#....#
#########	##O#.##.#
...#....#	#.O#.##.#
##.#.##.#	#.O#.OY.#
#..#.##.#	#.OOOO..#
#..#..Y.#	#########
#.......#	
#########	

Note that the time limit will be in the order of seconds. We expect your program to be able to solve large test cases very quickly. If your program takes several seconds to solve the sample test cases above, then you should not be surprised if your submission exceeds the time limit.

E Egg Enumeration

In recent weeks, the Egyptian embassy has endured an enlarged number of egg encounters, as more and more vandals and activists decided to display their disdain by discharging eggs towards the embassy. The staff has decided that something should be done, so now they are planning to report these acts of vandalism to the police. Before doing so, they want to know how many eggs have been thrown.

As you can imagine, the front lawn is a big mess, so counting individual eggs is virtually impossible. The consular staff therefore purchased the so called *Egg Enumeration Equipment*, often referred to as the EEE-scanner. This type of scanner can recognise individual egg whites as well as the yellow yolks, allowing the consular staff to estimate the number of eggs on their lawn. As the EEE-scanner is being moved across the front lawn, it outputs a stream of data. Your task is to analyse the data and determine the minimum number of eggs that must have been thrown towards the embassy.

Input

The input starts with a single integer T: the number of test cases. Then for each test case, a single line follows, which contains at least 1 and at most 10 000 characters. Every single character is one of the following:

- A capital letter `W` denotes an egg white turning up on the scanner;
- A capital letter `Y` denotes an egg yolk;
- A lowercase `n` is sent when the scanner sees nothing.

Advanced technology within the scanner hardware makes sure that no egg white or yolk is detected twice (or more). Of course, it is safe to assume that every egg contains exactly one egg white and exactly one yolk.

Output

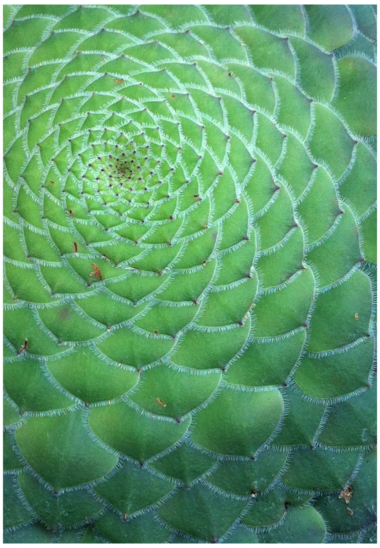
Based on the data from the EEE-scanner, determine the minimum number of eggs that have been thrown towards the embassy.

Sample input and output

Input	Output
`3`	`1`
`nnYnnnnWnnnnn`	`10`
`YYYYYYYYYY`	`5`
`nnWYWnnnYnWnnYnnWnnWnYnnnY`	

This is not a blank page

F Fibonacci



The fibonacci sequence occurs all over nature, as well as in various areas of mathematics. It is closely related to the golden ratio, and it has some other surprising mathematical properties as well. As we all know, the Fibonacci sequence is defined by

$$F_n = \begin{cases} 0, & \text{if } n = 0; \\ 1, & \text{if } n = 1; \\ F_{n-2} + F_{n-1}, & \text{if } n \geq 2. \end{cases}$$

The value of F_n is the sum of F_{n-2} and F_{n-1}. This can easily be generalised for larger sums. The generalised Fibonacci sequence is defined by

$$G_n = \begin{cases} n, & \text{if } 0 \leq n < m; \\ G_{n-m} + G_{n-m+1} + \ldots + G_{n-1}, & \text{if } n \geq m; \end{cases}$$

for some fixed positive integer m. Note that $G_n = F_n$ holds for all n if we choose $m = 2$. Your task is to calculate the value of G_n for given values of m and n. As this can grow rather large, you must reduce your answer modulo 123456789.

Input

The first line contains a single integer T: the number of test cases. This is followed by T lines, one for each test case, containing the integers n and m. These will satisfy $0 \leq n \leq 10^7$ and $1 \leq m \leq 10^7$.

Output

For each test case, output one line with a single integer: the value of G_n modulo 123456789. No answer greater than or equal to 123456789 or strictly smaller than 0 will be accepted.

Sample input and output

Input	Output

`9`
`0 2`
`1 2`
`2 2`
`3 2`
`4 2`
`5 2`
`6 2`
`100 100`
`1234567 123456`
	`0`
	`1`
	`1`
	`2`
	`3`
	`5`
	`8`
	`4950`
	`62516297`

If you need one or several large arrays, the best way to do that is to create a global, static array. In other words: create an array `int arr[LEN];` outside the main function.

This is not a blank page