Problem A
Cloud System

Time Limit: **3 seconds**

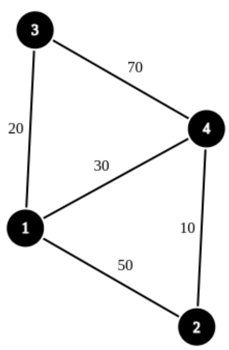
Mem limit: **512 Megabytes**

Ho Chi Minh University of Science has just built a new cloud computing system to run experiments related to deep-learning.

The system consists of N servers (numbered 1 to N). Each pair of servers are connected by at most 1 cable (possibly 0).

The cable connecting server u and server v has the transmitting capacity of $C_{u,v}$ megabits per nanosecond.

Let's define $F_{u,v}$ the data transfer rate between server u and server v. To transfer data from server u to server v, the data can be splitted into multi parts and transferred via multiple routes.



For example, to transfer data from server 1 to server 4, data can be split into 3 parts and transferred via 3 following routes:

- 1 - 4 (30 Mb per nanosecond)
- 1 - 3 - 4 (20 Mb per nanosecond)
- 1 - 2 - 4 (10 Mb per nanosecond)

so the transfer rate from 1 to 4 is 60 Mb per nanosecond or $F_{1,4} = 60$.

You are given the configuration of the network and the capacity of the cables, your task is to compute F.

Input

The input starts with T - the number of test cases. Then T test cases follow.

Each test case starts with N the number of servers in the network ($N \le 200$).

In the next N lines, each line contains N integers $C_{u,v}$ $(0 <= Cuv <= 10000)$.

It is guaranteed that $C_{u,v} = C_{v,u}$ and $C_{u,u} = 0$.

Note: The sum of N in the input does not exceeded 1000.

Output

For the t^{th} test case, print "`Case #t:`". The next N lines, print the $N \times N$ matrix F.

Sample input	Sample output
2 4 0 50 20 30 50 0 0 10 20 0 0 70 30 10 70 0 4 0 10 0 0 10 0 0 0 0 0 0 10 0 0 10 0	Case #1: 0 60 60 60 60 0 60 60 60 60 0 90 60 60 90 0 Case #2: 0 10 0 0 10 0 0 0 0 0 0 10 0 0 10 0

Problem B
Planning Tree

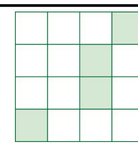
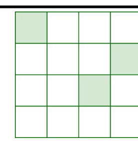
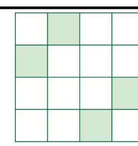
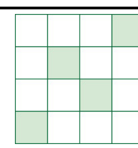
Time Limit: **1 second**
Mem limit: **512 Megabytes**

A new beautiful park has just been built in Hanoi. Harry is now responsible for planting trees in the park.

The park can be imagined as a grid with N rows and N columns. The rows are numbered from 1 to N from top to bottom. The columns are numbered from 1 to N from left to right. Let's denote the cell at i^{th} row and j^{th} column by (i, j).

Harry needs to plant exactly N trees, such that in each row and in each column, there is exactly one tree.

The entrance of the park is at cell $(1, 1)$, and the exit of the park is at cell (N, N). So Harry can not plant trees at these 2 cells. Furthermore, there must exist a path from the entrance to exit of the park such that visitors only need to move right or down. Obviously, the visitors cannot walk through trees.

The following park is not valid, because column 3 has 2 trees	
The following park is not valid, because cell $(1, 1)$ is not empty, and there are only 3 trees	
The following park is not valid, because it is not possible to go from $(1, 1)$ to $(4, 4)$ moving only right or down	
The following park is valid	

Please help Harry count how many valid parks are there. Two parks are considered different if there is at least one tree at different positions in the two parks.

Input

The first line of input contains a single integer T ($T \leq 10^5$) - the number of test cases.

T lines follow, each line contains a single integer N ($1 \leq N \leq 10^7$).

Output

Print exactly T lines, each line contains the number of ways to plant trees, satisfying all the given conditions. As the result can be very large, please prints the answer modulo $10^9 + 7$.

Sample Input / Sample Output

Sample Input	Sample Output
3 1 2 3 4	0 0 2 12

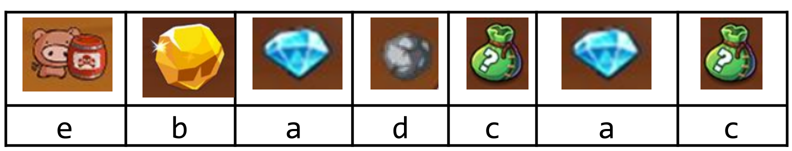
Problem C
Gold Miner

Time Limit: **3 seconds**
Mem limit: **512 Megabytes**

Mr. Phidang is a gold miner and he is at a gold mine at the moment. He has a powerful mining machine but it seems heavy. Mr. Phidang is going to use the machine to mine gold or other precious minerals (e,g. diamonds, silver, etc.).

There are many types of minerals in the gold mine and Mr. Phidang has his own preferences over them. The preference of each type is represented by a lowercase Latin letter and the preference rank is considered by its alphabetical order. For example, Mr. Phidang's most preferable type of mineral will be '`a`', the '`b`' type is a little bit less preferred, '`c`' type is for defining even worse, and so on.

The gold mine can be described as a continuous minable place on a line. The following figure describes the gold mine "`ebadcab`".



Mr. Phidang can start mining at any position that he wants. However, after started mining, he can only move to two adjacent slots (left or right) to continue mining since the current slot needs a unit of time to be refilled, and his mining machine is too heavy so he cannot move far away. Given that the amount of mineral at each slot is unlimited, Mr. Phidang can move back to the previous slots to mine.

Mr. Phidang is very greedy so he always wants to **maximize his preference of the minerals he earns at each step.** This also means that Mr. Phidang wants to choose the string that has the smallest alphabetical order.

Please help Mr. Phidang find the best starting place and the best way to achieve after a certain number of steps. For example with the gold mine as "`ebadcac`", the minerals Mr. Phidang earns after the 5 steps is "`ababa`", which is the best way for his preference.

Input

The first line contains the string s of lowercase Latin letters – the gold mine. The number of minable slots is not less than 2 and not greater than 100.

The second line contains one integer k $\left(1 \le k \le 10^4\right)$ – the number of mining steps.

Output

Print the sequence of letters without spaces – the values of minerals in order Mr. Phidang is going to mine.

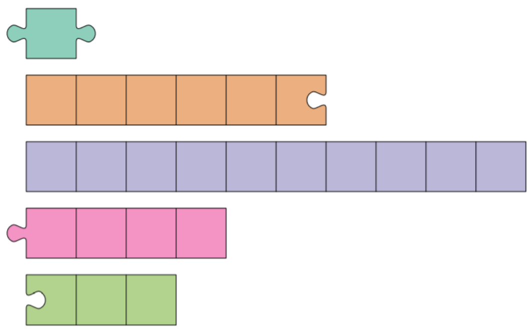
Sample Input / Sample Output

Sample Input	Sample Output
ebadcac 5	ababa
ccc 6	cccccc

Problem D
Puzzle Pieces

Time Limit: **2 seconds**
Mem limit: **512 Megabytes**

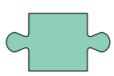
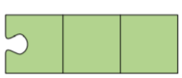
You are given n pieces of $1 \times a$ puzzle (one dimensional puzzle) as shown in the figure below.



Each piece has two heads which can be one of three types: "**in**", "**out**", or "**none**".

- The "**in**" type means that the border of the head is **concave**.
- The **"out"** type describe a **convex** border.
- The "**none**" type denotes a **straight** border.

For example,

	This piece has two **"out"** heads, or two **convex** heads.
	This piece has an **"in"** head on the left and a **"none"** head on the other side.

There are a few rules for you:

1. You cannot reverse these pieces, in other words, you cannot swap left and right borders of any piece.
2. Any "**in**" head can be connected with any **"out"** head and vice versa.
3. You cannot connect pieces that have **"none"** heads.

You have to connect many pieces (possibly one), one after another in order to achieve a single large piece of length L. Both heads of this combined piece must be **"none"** type. You wonder how many different sets of pieces, that you can build up the large piece of length L, using **all** the pieces in the set. Because the number of different sets could be large, you have to calculate it modulo $10^9 + 7$.

Note: You should count the number of sets of pieces, not the number of ways of connecting them.

Input

The first line contains two integers n and L – the number of puzzle pieces and desire length of the large piece ($1 \le n \le 300$, $1 \le L \le 300$).

The following n lines contain a description of the pieces. Each line contains an integer and two strings a_i, l_i and r_i — the length of the piece, type of its left head, and type of its right head, accordingly ($1 \le a_i \le L$; l_i, r_i $\in$ {"`in`", "`out`", "`none`"}). String "`in`" denotes concave border, "`out`" — convex, "`none`" — straight

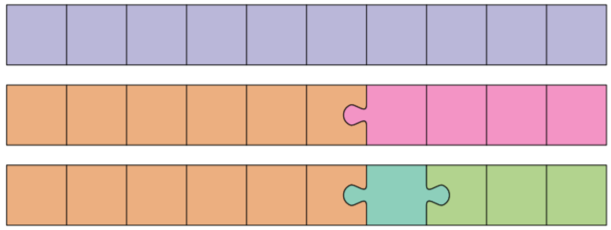
Output

Output a single integer — the number of sets of pieces, such that you can build desired the large piece using the given pieces, modulo $10^9 + 7$.

Sample Input	Sample Output
5 10 1 out out 6 none in 10 none none 4 out none 3 in none	3
4 5 1 none out 1 in out 2 in out 1 in none	1

Explain

The following figure explains the first sample test. There are three sets of pieces as illustrated in the figure.



Problem E
Martian Programmer

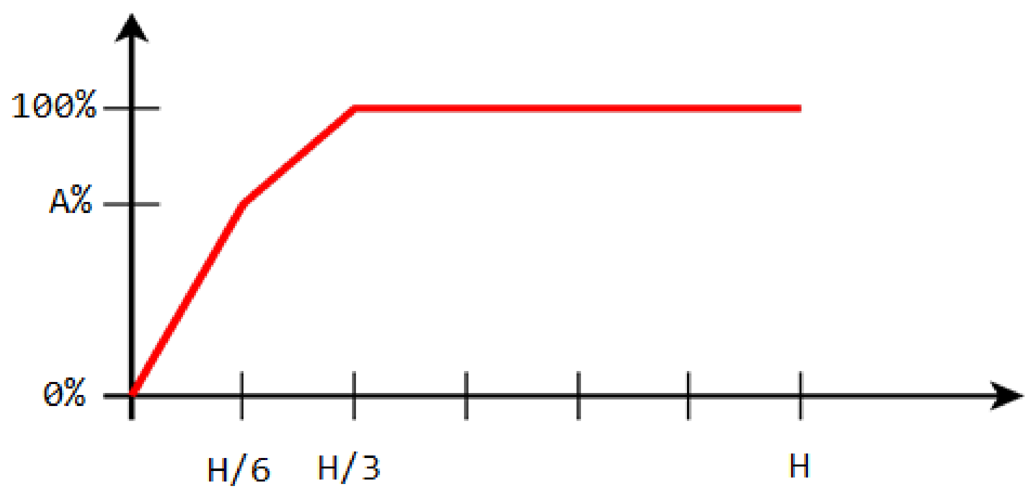
Time Limit: **1 second**

Mem limit: **512 Megabytes**

Mr. Phidang is an insane programmer coming from Mars, where there are exactly H hours a day. He has just read a scientific paper "Relatives between work and sleep". According to this research, all the time that you are not working, you are sleeping. Additionally, the research states and proves the following results:

1. If a person sleep 0 hours a day, then his performance will be 0%
2. If a person sleep $H/3$ hours a day, then his performance will be 100%
3. If a person sleep $H/6$ hours a day, then his performance will be A%
4. In case a person sleep from H/6 to H/3 hours a day, his performance increases linearly from A% to 100%
5. In case a person sleep from H/6 to 0 hours a day, his performance decreases linearly from A% to 0%

Mr. Phidang visualizes the information by drawing the following graph.



Mr. Phidang believes that the amount of programming work he do each day is equal to the product of his working time and his performance. Thus, the problem is that if he sleeps more, he works less, and if he sleep less, he has less performance.

Mr. Phidang wants to get back to programming as soon as possible. Please help him determine the maximum daily amount of work that he can do with the optimal choice of sleep time.

Input

The input contains two integers A and H – the performance as a percentage if the sleep time is H/6, and the number of hours per day on Mars ($0 \le A \le 100,\ 1 \le H \le 10^5$).

Output

The output should contains a single real number – the maximum daily amount of work that can be done by Mr. Phidang. The relative error does not exceed 10^{-6}.

Sample Input	Sample Output
75 24	1600.00000000
100 24	2000.00000000
77 123	8214.26086957

Problem F
Expected Value

Time Limit: **1 second**

Mem limit: **256 Megabytes**

Having a permutation $p = (p_1, p_2, \dots p_N)$ of the first N positive integers, let's define:

- $g_i(p)$ equals the greatest common divisor of the first i element of p $(1 \le i \le N)$.
- $f(p)$ equals the number of distinct integers in the array g.

For example, if $p = (2, 4, 6, 3, 1, 5)$ then

- $g_1 = GCD(2) = 2$
- $g_2 = GCD(2, 4) = 2$
- $g_3 = GCD(2, 4, 6) = 2$
- $g_4 = GCD(2, 4, 6, 3) = 1$
- $g_5 = GCD(2, 4, 6, 3, 1) = 1$
- $g_6 = GCD(2, 4, 6, 3, 1, 5) = 1$

Thus, $f(p)$ equals 2.

Given an integer N, we generate a random permutation p of size N (uniform random), your task is to calculate the expected value of $f(p)$.

Input

The input contains only one integer $(1 \le N \le 200,000)$.

Output

You should print the expected value of $f(p)$ modulo $10^9 + 7$.

Formally, let $M = 10^9 + 7$, it can be shown that the answer can be expressed as an irreducible fraction u / v where u and v are integers and $v != 0 \ (mod\ M)$. You should output the integer equal to $u * v - 1 \ (mod\ M)$. In other words, output such an integer x that $0 \le x < M$ and $x * v = u \ (mod\ M)$.

Sample input

Sample output

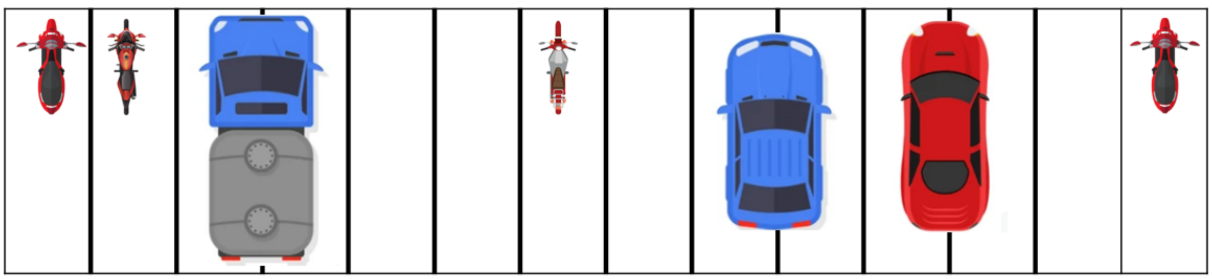
Sample input	Sample output
2	500000005

Problem G
Parking

Time Limit: **1 second**

Mem limit: **256 Megabytes**

Today, Mr. Tipu has a thesis defense at Ho Chi Minh University of Science, so he has driven his car to the University campus. The University parking lot is a line of n **parking zone**. Each zone has one meter in width, which fits one motorcycle. A car occupies exactly two adjacent zones of total width of two meters. The figure below illustrates the University parking lot.



Mr. Tipu wants to park his car in the parking lot, but some of the parking zones are occupied by other vehicles. Additionally, there are **a queue of k vehicles** that are trying to enter the parking lot in front of Mr. Tipu. The vehicles in the queue when entering the parking lot will occupy any empty zones: **one for a motorcycle and two adjacent zones for a single car**. In case there is no space left, all the current vehicles in the queue leave the University and not occupying any zone.

Because the thesis defense time is approaching, Mr. Tipu is in hurry. Fortunately, the people in the queue is very nice and they offer Mr. Tipu to enter the parking lot before them. Mr. Tipu doesn't want to take advantage of that so he wonder how many vehicles in front of him that he would pass to enter the parking lot on time, given that he will be late definitely in case there is no parking zones left that fit his car in the parking lot.

Note that the people in the queue may choose their parking zone for their vehicles imperfectly that leave no place for Mr. Tipu's car.

Given a parking lot of n parking zones, some of which are occupied, and a queue of k vehicles (could be motorcycles or cars) apart from Mr. Tipu's car, please help Mr. Tipu identify for each i from 0 to k, will Mr. Tipu be able to park his car, in case he enters the parking lot after the first i vehicles in the queue.

Input

There are several tests in each test case.

The first line contains a single integer t – the number of tests in the input ($1 \le t \le 50\,000$).

Each test consists of two lines.

The first of these two lines is a string of n characters ($1 \le n \le 10^5$), which can be '`.`' or '`X`'. The i^{th} character denotes the status of the i^{th} zone: '`.`' for empty and '`X`' for occupied.

The second of the two lines consist of k characters ($1 \le k \le n$), which can be '`M`' or '`C`', denoting motorcycle or car, respectively. The first character represents the first vehicle at the head of the queue and the last character is at the tail.

Output

The output contains t lines. Each line contains $k + 1$ characters, which can be '`Y`' or '`N`'. For each i from 0 to k print '`Y`', if Mr. Tipu can find a parking zone for his car, entering after the first i vehicles in the queue, no matter which zones they occupy, and print '`N`' otherwise.

Sample input	Sample output
1 XXXX..X.XXXX.X MM	YNN
3 ...... MMMC .X..X. MMM ...... CCCM	YYYNN YNNN YYNNN

Explanation

In the first sample, Mr. Tipu can park his car in the two empty parking zones in case he enter the parking lot first. Otherwise, the first motorcycle in the queue occupies one of the empty zones, so he won't be able to park his car.

In the test of the second sample, first three motorcycles could park in the following way: "`M.M.M.`", then Mr. Tipu could not be able to fine a two adjacent parking zones to park his car.