HKUST Local Contest Fall 2011
September 11, 2011

Contest Time: 1:00pm - 5:30pm

Letter	Time limit	Memory limit	Name
A	2 sec	64MB	Repeating Substrings
B	10 sec	64MB	Swimming Pool
C	2 sec	64MB	Rain Man's Problem
D	30 sec	64MB	The Great Escape
E	60 sec	64MB	The Hashing Problem
F	30 sec	64MB	Increasing Number

Contest Organizer:
Prof. Ke Yi
Mr. Derek Hao Hu

Problemsetter:
Mr. Derek Hao Hu
Mr. Yin Zhu

Judge:
Mr. Derek Hao Hu

Contest Rules and Regulations:

1. **This contest is an individual contest. Discussions between contestants are strictly prohibited.** Sanctions will be imposed on contestants if they are found to have violated the regulations governing integrity and honesty.

2. In this contest, **the contestants are given six programming problems**. The goal is to solve as many problems as possible. For those who solve the same number of problems, the one with lower score wins. (The scoring system will be explained below.)

3. **The programming language to be used in this contest is C/C++.** The contestants use PC^2 to submit their source codes to the judge and the source codes are compiled by Visual Studio.

4. **The contestant should read the input and write the output via standard I/O.** The contestants can assume that all test cases are of the format as stated in the problem statements. i.e. No exception handling is needed.

5. The correctness of each submission is judged by inputting test cases into the submitted program. The submission is regarded as correct if its outputs match completely with the model outputs. The submission is judged as correct or wrong. **No partial credit is given.**

6. The contestants can re-submit another source code after previous wrong submissions.

7. **All programs should not run for more than the time limit specified in the problem** (in most cases a "correct" implementation will run far less than the time limit we provide).

8. **The contestants are ranked firstly by the number of problems solved, and secondly the total time spent on solving the problems.** Time spent on solving one problem is the time between the start of contest and the submission of the correct implementation of that problem. For each problem you solved, a penalty of 20 minutes will be added to your score for each wrong submission of that problem.

9. **The contestants are allowed to bring any hard copies of books, notes, references, dictionaries and sketch papers to the contest site.** Electronic devices are forbidden.

Problem A. Repeating Substrings

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	2 seconds
Memory limit:	64 megabytes

You are given a string S with up to 50 characters, you are required to find the length of the longest substring that appears at least twice (non-overlapping and case sensitive) in S. If no such string exists in S, output 0.

Input

The first line of the input is an integer T, which indicates the number of test cases. T test cases follow. Each test case contains a string up to 50 characters.

Output

You should output T lines exactly. Each line contains the length of the longest substring appearing at least twice in the given string.

Example

Standard Input	Standard Output
5 ABCDEXXXYYYZZZABCDEZZZYYYXXX abcdabcdabcdabCD abcdefghijabcdefghij againANDagainANDagainANDagain abcdABCD	5 6 9 13 0

Problem B. Swimming Pool

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	10 seconds
Memory limit:	64 megabytes

A long time ago the construction of swimming pools was done thus: first, a rectangular area of length m meters and width n meters was designated. This area was then divided to 1×1 meter sized squares, thus obtaining $m \times n$ cells. Then either a perfectly cut granite cube with side length equal to 1, or a tower of these perfect cubes was placed in each cell. The cubes were so perfectly made that it was sufficient for two cubes to have their edges next to each other so that this place would be hermetically sealed (water would not flow through). When water was flowing, such a "set of cubes"could only hold a limited amount of water. The task is to write a program, which would determine the maximum possible volume of water in cubic meters that can be gathered in this pool.

Input

The first line of the input is an integer T, which indicates the number of test cases.

T test cases follow. Each test case contains several lines. The first line of the test case contains two positive integers m and n, which are the lengths of sides of the rectangular area in meters ($0 < m, n \leq 100$). The next of the m lines in the file contain each n natural numbers. The j-th number in the $i+1$-st line of the file shows the height of the cube tower located in the i-th row of the j-th column (or number of cubes positioned there). It is given that there is at least one cube in each cell and that the height of the tower does not exceed 10000 in any of the cells.

Output

You should output T lines exactly. Each line of the output must contain one positive integer - the maximum possible volume of water in cubic meters that can be stored in the given swimming pool.

Example

Standard Input	Standard Output
1 3 6 3 3 4 5 4 2 3 1 3 2 1 8 6 3 1 4 3 1	5

Problem C. Rain Man's Problem

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	5 seconds
Memory limit:	64 megabytes

Rain Man, the Oscar-winning film in 1998, stars *Dustin Hoffman* as Raymond Babbit, a man with autism, but also superb recall and extreme skill in mathematics. Intrigued by the fascinating skills of him, Derek has a test for you, to see if you can also have the ability of Rain Man's mathematical skills. The problem is quite simple, calculate the k-th digit in x^y. A value of 0 for k means the rightmost digit and a value of 1 means the digit next to the rightmost digit, etc. As a preliminary test, Derek has decided to make the problem as simple as possible.

Input

The first line of the input is an integer T, which indicates the number of test cases.

T lines follow, each line contains three integers x, y and k. $(0 \le x \le 10000, 0 \le y \le 10000, 0 \le k \le 4)$

Output

You should output T lines exactly. Each line outputs the k-th digit of x^y. If there is no k-th digit in x^y, output -1.

Example

Standard Input	Standard Output
4 2 10 1 2 10 4 2 1000 0 9999 10000 4	2 -1 6 0

Problem D. The Great Escape

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	30 seconds
Memory limit:	64 megabytes

Farmer John has N cows numbered from 0 to $N-1$. The cows, though well fed, have decided to run away from the farm one night. A passenger walking-by had observed this and told Farmer John that K cows ran away from the farm and that the sum of the numbers of the escaped cows was divisible by N. Now your task is to help Farmer John by counting the number of possible sets of escaped cows. Since this number may be very big, please output it modulo 1,000,000,007.

Input

The first line of the input contains an integer T, indicating the number of test cases. T lines follow, each line contains two integers N and K. ($1 \le N \le 10^9, 1 \le K \le 1000, K < N$)

Output

You should output T lines where each line indicates the number of possible sets of escaped cows modulo 1,000,000,007.

Example

Standard Input	Standard Output
5 7 4 1 1 58 4 1000 47 1000000000 1000	5 1 7322 219736903 666683069

Problem E. The Hashing Problem

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	60 seconds
Memory limit:	64 megabytes

Being a computer science student you must have known hashing as one of the most basic data structure you've learnt. In this problem, we are given a very simple hashing function $h(y) = ay + b \mod m$ which maps each integer to some integer between 0 and $m - 1$. Now you are given x, n, c, d and you are required to count how many of the hash values $h(x), h(x+1), \ldots, h(x+n)$ lie within the interval of $[c, d]$.

Input

The first line of the input contains an integer T, indicating the number of test cases. T lines follow, where each line gives the values a, b, x, n, c, d, m for each test case. All given values are non-negative. Besides, $1 \leq m \leq 10^{15}$, $c \leq d < m$, $a, b < m$, $x + n \leq 10^{15}$ and $a \cdot (x + n) + b \leq 10^{15}$.

Output

For each test case output the number of i, $0 \leq i \leq n$, such that $c \leq a \cdot (x + i) + b \mod m \leq d$ in that test case.

Example

Standard Input	Standard Output
2 2 3 1 3 0 1 7 1 0 0 8 0 8 9	1 9

Problem F. Increasing Number

Input file:	`Standard Input`
Output file:	`Standard Output`
Time limit:	30 seconds
Memory limit:	64 megabytes

A positive integer is called an increasing number if its digits are in non-descending order from left to right. For example, 1234 and 111 are increasing numbers while 324 and 2011 are not. You are required to calculate the number of increasing numbers satisfying both of the following conditions:

- The number contains exactly D digits in its decimal notation.
- The number is divisible by Q.

Since this number may be very big, output this number modulo 1,000,000,007.

Input

The first line of the input contains an integer T, indicating the number of test cases. T test cases follow, each test case contain two numbers, D and Q. ($1 \leq D \leq 10^{18}, 1 \leq Q \leq 500$).

Output

For each test case, output the number of increasing numbers satisfying the two conditions mentioned above.

Example

Standard Input	Standard Output
4 2 12 3 111 452 10 6 58	4 9 0 38