2017 中国大学生程序设计竞赛
女生专场

2017 CCPC-WFINAL

2017 年 5 月 6 日

Problem 1001. Automatic Judge

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 seconds
Memory limit:	128 megabytes

Welcome to HDU to take part in the second CCPC girls’ competition!

A new automatic judge system is used for this competition. During the five-hour contest time, you can submit your code to the system, then the judge will reply you. Here is a list of the judge's replies and their meaning:

- Accepted (AC) : Yes, your program is correct. You did a good job!
- Presentation Error (PE) : Your program's output format is not exactly the same as required by the problem, although the output is correct. This usually means the existence of omitted or extra blank characters (white spaces, tab characters and/or new line characters) between any two non-blank characters, and/or blank lines (a line consisting of only blank characters) between any two non-blank lines. Trailing blank characters at the end of each line and trailing blank lines at the of output are not considered format errors. Check the output for spaces, blank lines, etc. against the problem's output specification.
- Wrong Answer (WA) : Correct solution not reached for the inputs. The inputs and outputs that we use to test the programs are not public (it is recomendable to get accustomed to a true contest dynamic :-)
- Runtime Error (RE) : Your program failed during the execution and you will receive the hints for the reasons.
- Time Limit Exceeded (TLE) : Your program tried to run during too much time.
- Memory Limit Exceeded (MLE) : Your program tried to use more memory than the judge default settings.
- Output Limit Exceeded (OLE) : Your program tried to write too much information. This usually occurs if it goes into a infinite loop.
- Compilation Error (CE) : The compiler fails to compile your program. Warning messages are not considered errors. Click on the judge's reply to see the warning and error messages produced by the compiler.

For each submission, if it is the first time that the judge returns “AC” on this problem, then it means you have passed this problem, and the current time will be added to the penalty of your team. In addition, every time you pass a problem, each unsuccessful try for that problem before is counted as 20 minutes penalty, it should also be added to the penalty of your team.

Now given the number of problems in the contest and the submission records of a team. Please write a program to calculate the number of problems the team passed and their penalty.

Input

The first line of the input contains an integer $T(1 \leq T \leq 20)$, denoting the number of test cases.

In each test case, there are two integers $n(1 \leq n \leq 13)$ and $m(1 \leq m \leq 100)$ in the first line, denoting the number of problems and the number of submissions of a team. Problems are labeled by 1001, 1002, ..., $1000 + n$.

In the following m lines, each line contains an integer $x(1001 \leq x \leq 1000 + n)$ and two strings $t(00:00 \leq t \leq 05:00)$ and s, denoting the team submits problem x at time t, and the result is s. t is in the format of HH:MM, while s is in the set {AC, PE, WA, RE, TLE, MLE, OLE}. The team is so cautious that they never submit a CE code. It is guaranteed that all the t in the input is in ascending order and every t is unique.

Output

For each test case, print a single line containing two integers A and B, denoting the number of problems the team passed and the penalty.

Examples

`stdin`	`stdout`
1 3 5 1002 00:02 AC 1003 00:05 WA 1003 00:06 WA 1003 00:07 AC 1002 04:59 AC	2 49

Problem 1002. Building Shops

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 seconds
Memory limit:	128 megabytes

HDU’ s n classrooms are on a line ,which can be considered as a number line. Each classroom has a coordinate. Now Little Q wants to build several candy shops in these n classrooms.

The total cost consists of two parts. Building a candy shop at classroom i would have some cost c_i. For every classroom P without any candy shop, then the distance between P and the rightmost classroom with a candy shop on P's left side would be included in the cost too. Obviously, if there is a classroom without any candy shop, there must be a candy shop on its left side.

Now Little Q wants to know how to build the candy shops with the minimal cost. Please write a program to help him.

Input

The input contains several test cases, no more than 10 test cases.

In each test case, the first line contains an integer $n(1 \leq n \leq 3000)$, denoting the number of the classrooms.

In the following n lines, each line contains two integers $x_i, c_i(-10^9 \leq x_i, c_i \leq 10^9)$, denoting the coordinate of the i-th classroom and the cost of building a candy shop in it.

There are no two classrooms having same coordinate.

Output

For each test case, print a single line containing an integer, denoting the minimal cost.

Examples

stdin	stdout

3
1 2
2 3
3 4
4
1 7
3 1
5 10
6 1
	5
	11

Problem 1003. Coprime Sequence

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 seconds
Memory limit:	128 megabytes

Do you know what is called “Coprime Sequence”? That is a sequence consists of n positive integers, and the GCD (Greatest Common Divisor) of them is equal to 1.

“Coprime Sequence” is easy to find because of its restriction. But we can try to maximize the GCD of these integers by removing exactly one integer. Now given a sequence, please maximize the GCD of its elements.

Input

The first line of the input contains an integer $T(1 \leq T \leq 10)$, denoting the number of test cases.

In each test case, there is an integer $n(3 \leq n \leq 100000)$ in the first line, denoting the number of integers in the sequence.

Then the following line consists of n integers $a_1, a_2, ..., a_n (1 \leq a_i \leq 10^9)$, denoting the elements in the sequence.

Output

For each test case, print a single line containing a single integer, denoting the maximum GCD.

Examples

stdin	stdout

3
3
1 1 1
5
2 2 2 3 2
4
1 2 4 8
	1
	2
	2

Problem 1004. Deleting Edges

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 seconds
Memory limit:	128 megabytes

Little Q is crazy about graph theory, and now he creates a game about graphs and trees.

There is a bi-directional graph with n nodes, labeled from 0 to $n-1$. Every edge has its length, which is a positive integer ranged from 1 to 9.

Now, Little Q wants to delete some edges (or delete nothing) in the graph to get a new graph, which satisfies the following requirements:

(1) The new graph is a tree with $n-1$ edges.

(2) For every vertice $v(0 < v < n)$, the distance between 0 and v on the tree is equal to the length of shortest path from 0 to v in the original graph.

Little Q wonders the number of ways to delete edges to get such a satisfied graph. If there exists an edge between two nodes i and j, while in another graph there isn't such edge, then we regard the two graphs different.

Since the answer may be very large, please print the answer modulo 10^9+7.

Input

The input contains several test cases, no more than 10 test cases.

In each test case, the first line contains an integer $n(1 \leq n \leq 50)$, denoting the number of nodes in the graph.

In the following n lines, every line contains a string with n characters. These strings describes the adjacency matrix of the graph. Suppose the j-th number of the i-th line is $c(0 \leq c \leq 9)$, if c is a positive integer, there is an edge between i and j with length of c, if $c=0$, then there isn't any edge between i and j.

The input data ensure that the i-th number of the i-th line is always 0, and the j-th number of the i-th line is always equal to the i-th number of the j-th line.

Output

For each test case, print a single line containing a single integer, denoting the answer modulo 10^9+7.

Examples

stdin	stdout
2 01 10 4 0123 1012 2101 3210	1 6

Problem 1005. Easy Summation

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 seconds
Memory limit:	128 megabytes

You are encountered with a traditional problem concerning the sums of powers.

Given two integers n and k. Let $f(i) = i^k$, please evaluate the sum $f(1) + f(2) + ... + f(n)$. The problem is simple as it looks, apart from the value of n in this question is quite large.

Can you figure the answer out? Since the answer may be too large, please output the answer modulo $10^9 + 7$.

Input

The first line of the input contains an integer $T(1 \leq T \leq 20)$, denoting the number of test cases.

Each of the following T lines contains two integers $n(1 \leq n \leq 10000)$ and $k(0 \leq k \leq 5)$.

Output

For each test case, print a single line containing an integer modulo $10^9 + 7$.

Examples

stdin	stdout

3
2 5
4 2
4 1
	33
	30
	10

Problem 1006. Forgiveness

Input file:	`stdin`
Output file:	`stdout`
Time limit:	7 seconds
Memory limit:	128 megabytes

Little Q is now checking whether string A matches B. Two strings are considered matched if they have the same length, and there are no position i that A_i is different from B_i.

However, Little Q is a kind man, he forgives every person hurt him. What's more, he even forgives strings! He gives the string 3 opportunities, if there are no more than 3 positions i that A_i is different from B_i, then Little Q will also consider the two strings matched.

For a string S, $S[l,r]$ means the substring combined by $S_l, S_{l+1}, ..., S_r$. And the function $occ(A,B)$ returns the number of substrings in string B which matches A.

Little Q now has a long numeric 1-based string S, and his job is to deal with m operations:

- $+\ l\ r\ k$, for every positions from l to r, change S_i to $(S_i + k) \bmod 10$.
- $?\ l\ r\ T$, report $occ(T, S[l,r])$.

After lots of work, Little Q is very tired now, please write a program to help him deal with these operations.

Input

The first line of the input contains an integer $T(1 \leq T \leq 15)$, denoting the number of test cases.

In each test case, there are two integers $n(1 \leq n \leq 50000)$ and $m(1 \leq m \leq 50000)$ in the first line, denoting the length of string S and the number of operations.

The second line of the input contains a numeric string S with n integers, each number S_i is in the range of 0 to 9.

In the following m lines, each line describes an operation.

If it is a modification, then it is in the format of "$+\ l\ r\ k$", where $1 \leq l \leq r \leq n$ and $1 \leq k \leq 9$.

If it is a query, then it is in the format of "$?\ l\ r\ T$", where $1 \leq l \leq r \leq n$ and T is a numeric string composed of integers from 0 to 9.

It is guaranteed that $\sum |T| \leq 100000$ in each test case, and there are no more than 4 test cases satisfying $\min(n,m) > 1000$.

Output

For each query, print a single line with an integer, denoting the answer.

Examples

stdin	stdout
1 5 5 01234 ? 2 5 1234 ? 2 5 1777 ? 2 5 9999 + 1 5 5 ? 1 5 56789	1 1 0 1

Problem 1007. Graph Theory

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 seconds
Memory limit:	128 megabytes

Little Q loves playing with different kinds of graphs very much. One day he thought about an interesting category of graphs called “Cool Graph”, which are generated in the following way:

Let the set of vertices be $\{1, 2, 3, ..., n\}$. You have to consider every vertice from left to right (i.e. from vertice 2 to n). At vertice i, you must make one of the following two decisions:

(1) Add edges between this vertex and all the previous vertices (i.e. from vertex 1 to $i-1$).

(2) Not add any edge between this vertex and any of the previous vertices.

In the mathematical discipline of graph theory, a matching in a graph is a set of edges without common vertices. A perfect matching is a matching that each vertice is covered by an edge in the set.

Now Little Q is interested in checking whether a “Cool Graph” has perfect matching. Please write a program to help him.

Input

The first line of the input contains an integer $T(1 \leq T \leq 50)$, denoting the number of test cases.

In each test case, there is an integer $n(2 \leq n \leq 100000)$ in the first line, denoting the number of vertices of the graph.

The following line contains $n - 1$ integers $a_2, a_3, ..., a_n (1 \leq a_i \leq 2)$, denoting the decision on each vertice.

Output

For each test case, output a string in the first line. If the graph has perfect matching, output “Yes”, otherwise output “No”.

Examples

stdin	stdout
3	Yes
2	No
1	No
2	
2	
4	
1 1 2	

Problem 1008. Happy Necklace

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 seconds
Memory limit:	128 megabytes

Little Q wants to buy a necklace for his girlfriend. Necklaces are single strings composed of multiple red and blue beads.

Little Q desperately wants to impress his girlfriend, he knows that she will like the necklace only if for every prime length continuous subsequence in the necklace, the number of red beads is not less than the number of blue beads.

Now Little Q wants to buy a necklace with exactly n beads. He wants to know the number of different necklaces that can make his girlfriend happy. Please write a program to help Little Q. Since the answer may be very large, please print the answer modulo $10^9 + 7$.

Note: The necklace is a single string, **not a circle**.

Input

The first line of the input contains an integer $T(1 \leq T \leq 10000)$, denoting the number of test cases.

For each test case, there is a single line containing an integer $n(2 \leq n \leq 10^{18})$, denoting the number of beads on the necklace.

Output

For each test case, print a single line containing a single integer, denoting the answer modulo 10^9+7.

Examples

`stdin`	`stdout`
2 2 3	3 4

Problem 1009. Innumerable Ancestors

Input file:	stdin
Output file:	stdout
Time limit:	5 seconds
Memory limit:	128 megabytes

There is a tree having n nodes, labeled from 1 to n. The root of the tree is always 1, and the depth of a node p is the number of nodes on the shortest path between node p and the root.

In computer science, the Lowest Common Ancestor (LCA) of two nodes v and w in a tree is the lowest (i.e. deepest) node that has both v and w as descendants, where we define each node to be a descendant of itself (so if v has a direct connection from w, w is the lowest common ancestor).

You have to answer m queries. Each query gives two non-empty node sets A and B, there might be some nodes in both sets.

You should select one node x from set A, and one node y from set B, x and y can be the same node. Your goal is to maximize the depth of the LCA of x and y.

Please write a program to answer these queries.

Input

The input contains several test cases, no more than 5 test cases.

In each test case, the first line contains two integers $n(1 \leq n \leq 100000)$ and $m(1 \leq m \leq 100000)$, denoting the number of nodes and queries.

For the next $n-1$ lines,each line contians two integers a and b, denoting a bi-directional edge between node a and b.

Then there are $2m$ lines, every two lines describes one query.

For each query, the first line describes the set A.

The first integer $k(1 \leq k \leq n)$ denotes the number of nodes in set A, and the next k integers describing the nodes in set A. There might be some nodes appear multiple times in the set.

The second line describes the set B in the same format of set A.

It is guaranteed that $\sum k \leq 100000$ in each test case.

Output

For every query, print a number denoting the answer, which means the maximum depth of the LCA.

Examples

stdin	stdout

7 3
1 2
1 3
3 4
3 5
4 6
4 7
1 6
1 7
2 6 7
1 7
2 5 4
2 3 2
	3
	4
	2

Problem 1010. Judicious Strategy

Input file:	`stdin`
Output file:	`stdout`
Time limit:	1 seconds
Memory limit:	128 megabytes

Alice and Bob is now playing a game about strings.

There is a dictionary containing n words (words might be same). Alice choose a lowercase English letter arbitrarily first, but this letter should appear in at least one of these n words. Then Bob choose a lowercase English letter arbitrarily to add it before or after the letter Alice chose. So Bob gets a new string now. This new string should also be a substring (consecutive subsequence) of at least one strings in the dictionary. After that, it's Alice's turn. Alice should do the same thing, choosing a letter and add it before or after the current string, making a new string. At every moment, the string they made should always be a substring of at least one strings in the dictionary. The player who can't operate first lose the game and the other one win.

Besides, each player has a score. The score is calculated by the following rule:

If the string S is now made, the current player will get $score(S)$ points. It means that Alice will score in the first round, then Bob, then Alice...

$$score(S) = \left[\left(\sum_{i=1}^{|S|} value(S_i)\right) \times \max_{i=1}^{|S|} value(S_i)\right] + occ(S)$$

where

- $|S|$ means the length of S.
- $value(c)$ represents the value of letter c. The score of letter "a" is 1, "b" is 2, ..., "z" is 26.
- $occ(S)$ means the time that S occurs as a substring in the dictionary, each word is counted just once.

Alice and Bob will play with best strategy. That is to say, they will consider to win first and then maximize their score, after that they will consider to minimize the score of others.

Please determine who will win the game, and report the final scores they will earn during the whole game.

Input

The input contains several test cases, no more than 10 test cases.

In each test case, the first line contains an integer $n (1 \leq n \leq 30)$, denoting the number of words in the dictionary.

In the next n lines, each line contains a non-empty string $word_i$, denoting a word in the dictionary. The string is composed of lowercase English letters and its length will not exceed 30.

Output

For each test case, output a string in the first line. If Alice will win,output “Alice”, otherwise output “Bob”.

Then print two integers A and B in second line, denoting the final score of Alice and Bob.

Examples

stdin	stdout
2 aba abac 3 artem nik max	Bob 29 35 Alice 2403 1882