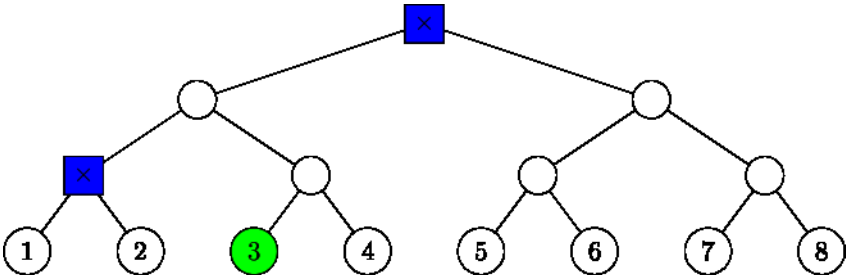
Problem A

Worst Locations

Two pandas A and B like each other. They have been placed in a bamboo jungle (which can be seen as a perfect binary tree graph of 2^N-1 vertices and 2^N-2 edges whose leaves are all of the same depth) at different locations. The jungle organizers (yes, such organizers exist), being a bit disorganized, only note down two integers X and Y as an indicator of the location of a panda which is translated as: that panda is now at a vertex whose distance is exactly Y vertices away from the leaf X. The leaves are numbered from 1 to 2^{N-1} from left to right. As you might have noticed, this indicator may correspond to more than one vertex. For example, the image below shows possible locations for N = 4, X = 3, Y = 3.



Back to our two pandas, the indicators of the locations of these two pandas are (X_A, Y_A) and (X_B, Y_B). One can shout to another such that if they are at most Z vertices away from each other, the other can hear that shout. The question is, given the height of the jungle layout, the location indicators of these two pandas and the strength of their shouts, is it possible that these two pandas **cannot** hear each other's shout?

Input

The first line of input contains an integer T (T ≤ 50,000) denoting the number of testcases. Each testcase is represented by a single line containing 6 space-separated integers N, X_A, Y_A, X_B, Y_B, and Z (1 ≤ N ≤ 31; 1 ≤ X_A, X_B ≤ 2^{N-1}; 0 ≤ Y_A, Y_B, Z ≤ 2 * N - 2) denoting the height of the perfect binary tree jungle layout, the location indicators of the two pandas, and the strength of their shouts.

Output

For each testcase, write "YES" to a single line of output if it is the case that these two pandas might not be able to hear each other's shout, and "NO" otherwise.

Sample Input	Output for Sample Input
2 4 3 3 3 3 2 4 3 3 3 3 1	NO YES

Problem B
Counting BST

Binary Search Tree (BST) is a rooted binary tree data structure which has following properties:

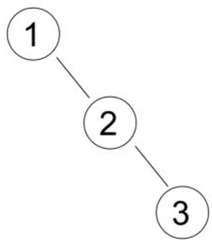
- Left subtree contains only nodes with value less than the node's value.
- Right subtree contains only nodes with value greater than the node's value.
- All values in the nodes are unique.
- Both left and right subtrees are also binary search tree recursively.

If there is a new node to be inserted, the following algorithm will be used:

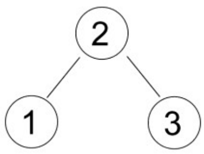
1. If the root is empty, then the new node becomes the root and quit, else continue to step 2.
2. Set the root as current node.
3. If the new node's value is less than current node's value:
   - If current node's left is empty, then set the new node as current node's left-child and quit.
   - else set current node's left-child as current node, and repeat step 3.
4. If the new node's value is greater than current node's value:
   - If current node's right is empty, then set the new node as current node's right-child and quit.
   - else set current node's right-child as current node, and repeat step 3.

BST structure depends on its data inserting sequence. Different sequence may yield a different structure though the data set is the same. For example:

Insert sequence: 1 2 3, the BST will be:

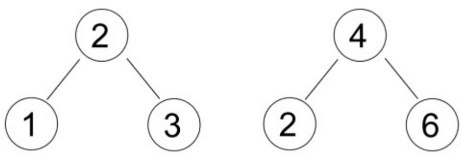


If the data is inserted with sequence: 2 1 3, the tree will be:



On the other hand, different data set may have a same BST structure. For example:

Insert sequence 2 1 3 will have the same BST structure with 4 6 2, and the tree will be:



Given N nodes BST, calculate how many distinct insert data sequence which result in the same BST structure, assuming that data are taken from range 1..M.

Input

The first line of input contains an integer T (T ≤ 100), the number of test cases. Each case begins with two integers N and M (1 ≤ N ≤ M ≤ 1,000), the number of nodes in BST and the maximum range respectively. The next line contains N integers A_i (1 ≤ A_i ≤ 1,000) the insert sequence that construct a BST.

Output

For each case, output an integer denoting the number of distinct insert data sequence which result in the same BST structure, assuming that data are taken from range 1..M. Modulo this number with 1,000,003.

Sample Input	Output for Sample Input
3 3 4 3 1 4 3 5 1 2 3 4 4 2 1 10 3	8 10 3

Explanation for the 1st sample input.

There are 8 insert sequences (data taken from 1..4) which have the same BST:

1. 2 1 3
2. 2 3 1
3. 2 1 4
4. 2 4 1
5. 3 1 4
6. 3 4 1
7. 3 2 4
8. 3 4 2

Problem C

Playing With Stones

You and your friend are playing a game in which you and your friend take turns removing stones from piles. Initially there are N piles with a_1, a_2, a_3, …, a_N number of stones. On each turn, a player must remove at least one stone from one pile but no more than half of the number of stones in that pile. The player who cannot make any moves is considered lost. For example, if there are three piles with 5, 1 and 2 stones, then the player can take 1 or 2 stones from first pile, no stone from second pile, and only 1 stone from third pile. Note that the player cannot take any stones from the second pile as 1 is more than half of 1 (the size of that pile). Assume that you and your friend play optimally and you play first, determine whether you have a winning move. You are said to have a winning move if after making that move, you can eventually win no matter what your friend does.

Input

The first line of input contains an integer T (T ≤ 100) denoting the number of testcases. Each testcase begins with an integer N (1 ≤ N ≤ 100) the number of piles. The next line contains N integers a_1, a_2, a_3, …, a_N ($1 \le ai \le 2 * 10^{18}$) the number of stones in each pile.

Output

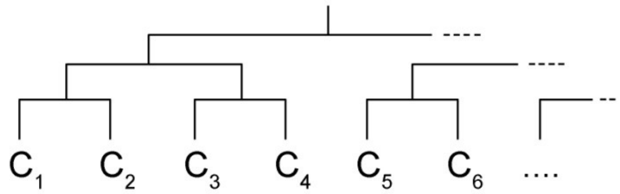
For each testcase, print "YES" (without quote) if you have a winning move, or "NO" (without quote) if you don't have a winning move.

Sample Input	Output for Sample Input
4 2 4 4 3 1 2 3 3 2 4 6 3 1 2 1	NO YES NO YES

Problem D

Arm Wrestling Tournament

As you might have heard, Mr. Kumis is holding an arm wrestling tournament. There are 2^N contestants who will participate in this tournament numbered from 1 to 2^N. The first contestant (C_1) will compete with the second contestant (C_2). C_3 will compete with the C_4, and so on. The winner of C_1 and C_2 will compete with the winner of C_3 and C_4. The winner of C_5 and C_6 will compete with the winner of C_7 and C_8, and so on (see the diagram below).



Each contestant initially has P_i strength. When two contestants wrestle, the stronger one will win and his strength will be reduced as much as his enemy's strength. However, before his next match, he has time to regain his strength and will recover at most K strength but his strength will not exceed his initial strength (P_i). If two contestants possess an equal strength then the contestant with smaller index will win.

Given the initial strength of all contestants, determine who will win the tournament and which contestant he will beat.

Input

The first line of input contains an integer T (T ≤ 100) denoting the number of testcases. Each testcase begins with two integer N (1 ≤ N ≤ 15) and K (0 ≤ K ≤ 1,000). The next line contains 2^N integers P_i (1 ≤ Pi ≤ 1,000) denoting the initial strength of i^{th} contestant for i = 1..2^N.

Output

For each testcase, print two lines. The first line contains an integer, the winner of the tournament. The second line contains N integers which are all contestants the winner beat based on match order. Each integer is separated by a single space.

Sample Input	Output for Sample Input
3 3 0 100 90 79 37 60 50 39 95 2 10 50 50 60 60 2 10 3 5 60 59	8 7 5 3 1 2 3 3 4 2

Problem E

Lightning Energy Report

The city of Thunder has many houses and they are powered by lightning. Two houses may be interconnected by a wire. The wires connect the houses such that there is exactly one path from one house to any other house in the city. Each house has a large battery that can store infinite electrical energy.

Everyday, several lightnings strike several houses. These lightnings are very unusual, when it strikes, it strikes two houses simultaneously: one house (A) with a red lightning and the other house (B) with a blue lightning. After the strikes, every house along the path from A to B (inclusive) will receive a certain amount of electrical energy which is then added to the house's battery.

The mayor of Thunder city wants a report on the stored electrical energy for every house at the end of the month and the owner of each house has to pay taxes for that. To prevent the owners report invalid energy stored in their house battery (because they want less tax to pay), you are asked to produce a correct report based on the observed lightnings of the current month. At the beginning of the month, the energy reading on the battery of each house is 0.

Input

The first line of input contains an integer T (T ≤ 10), the number of cases. Each case begins with an integer N (2 ≤ N ≤ 50,000), the number of houses in the Thunder city. The next N-1 lines contain the wire connections where each line will consist of two integers X and Y which means house X is connected with house Y. The house number is from 0 to N-1. The next line will contain a number Q (1 ≤ Q ≤ 50,000) which denotes the number of observed lightning strikes. The next Q lines describe the unusual lightnings happened during the month. Each line will consist of three integers A, B, C which tells that a red lightning strikes house A and a blue lightning strikes house B and the power transferred is C (at most 100) based on the reading of a special lightning instrument.

Output

For each case, output "`Case #X:`" (without quote) where X is the case number and N lines where each line is the electrical energy reading on the battery of each house from house 0 to house N-1 at the end of the month.

Sample Input	Output for Sample Input
1 9 0 1 1 2 2 3 2 4 2 7 7 8 7 6 6 5 5 1 4 10 3 5 3 0 8 5 1 6 10 4 4 100	Case #1: 5 25 28 3 110 3 13 18 5

Problem F

Transitive Closure

In almost all ICPC trainings, one of the basic items to be taught (or even better, assumed) is how to compute the transitive closure of a directed graph. Since this concept is so elementary, there is a professor that suggests to solve this problem using Warshall's algorithm. As one of the bonus tasks of this ICPC, we would like to examine whether the participants have indeed mastered this technique.

Given a directed graph G = <V, E> with N = |V| vertices ($2 \le N \le 2{,}500$) and M = |E| edges ($1 \le M \le 10{,}000$), its transitive closure is the binary relation R on V such that two vertices, A and B, are related in R if there is a directed path from A to B, where a directed path is a sequence of edges of the graph C_0C_1, C_1C_2, ..., $C_{K-1}C_K$ such that $A = C_0$ and $B = C_K$. Your task is to compute R.

Because R might be very large, we've decided to simplify the output. Calculate the number of ordered pairs of (X,Y) where X is not equal to Y such that there is a path from vertex X to vertex Y in graph G.

Input

The first line of input contains an integer T ($T \le 10$), denoting the number of testcases. The following lines describe each test case of the following format.

The first line of a testcase consists of two integers N and M. Each of the following M lines contains 2 integers, A and B, denoting that there is a directed edge from A to B. The vertices are numbered from 1 to N, and there will be no repeated edges within a description of a graph.

Output

For each case, output an integer denoting the number of ordered pairs of (X,Y) where X is not equal to Y such that there is a path from vertex X to vertex Y.

Sample Input	Output for Sample Input
2 3 2 1 2 2 3 4 4 1 2 2 3 2 4 4 1	3 9

Explanation for the 1st sample input.

The binary relation R is:

```
0 1 1
0 0 1
0 0 0
```

There are 3 ordered pairs of (X,Y) such that $R_{X,Y}$ is true.

Problem G

Just Sum It

Given the number of available digit of 1 to 9, sum all possible numbers generated from those digits.

For example,

Digit	1	2	3	4	5	6	7	8	9
Frequency	0	2	0	1	0	1	0	0	0

It means that we can use up to two digits of 2, one digit of 4 and one digit of 6. There are exactly 32 distinct numbers that can be constructed using the above digits: 2, 4, 6, 22, 24, 26, 42, 46, 62, 64, 224, 226, 242, 246, 262, 264, 422, 426, 462, 622, 624, 642, 2246, 2264, 2426, 2462, 2624, 2642, 4226, 4262, 4622, 6224, 6242, 6422. The sum of all those numbers is 51622.

Input

The first line of input contains an integer T (T ≤ 500) denoting the number of testcases. Each testcase contains nine integers P_i ($0 \le P_i \le 9$) denoting the number of i-th digit for i = 1..9.

Output

For each testcase, output in a single line the sum of all possible numbers generated from the available digits. Modulo the output with 1,000,000,007.

Sample Input	Output for Sample Input
3 0 0 1 0 1 0 0 0 0 0 2 0 1 0 1 0 0 0 1 1 1 1 1 1 1 1 1	96 51622 454976431

Problem H

Serial Numbers

In the factory of ACM Computerized Manufacturing Ltd. (ACM Ltd.), items are manufactured in assembly lines and are given imprinted serial numbers to distinguish them from one another. The serial numbers are generated by a special kind of simplistic multi-function counter machine which generates serial number sequentially in increasing order (1, 2, 3, and so on).

As the counter machine has multiple functions embedded in its simple design, some substrings are reserved as op-codes command which must not be presented in the machine's serial number output. For example if 23 is a command op-code, then 23 must not be presented in the substring of a serial number – assuming there is no other command op-code, the number sequence would go like this: 1, 2, 3, …, 21, 22, 24, 25, …, 121, 122, 124, 125, …, 228, 229, 240, 241, 242, ….

This result in serial number that is different from the actual production batch number recorded in the system. The system does not map the relationship between both numbers and/or record exact serial numbers for each item, this may cause problems when an individual item needs to be tracked (e.g. for a recall or inspection). Given the production batch number of an item, your task is to determine the serial number for that item.

Input

The first line of input contains an integer T (T ≤ 100) denoting the number of testcases. Each testcase begins with an integer K (1 ≤ K ≤ 10) the number of op-codes in the counter machine, followed by K integers (1 to 10 digits, with leading zeroes preserved) representing each op-codes. The second line contains an integer N (1 ≤ N ≤ 100) the number of requested production batch number, followed by N integers representing each production batch number.

Output

For each testcase, print N integers the serial numbers of corresponding production batch number. Two adjacent integers are separated by a single space. You may assume that these numbers are fit in signed 32-bit integer.

Sample Input	Output for Sample Input
4 1 4 2 3 5 1 2 3 4 5 6 2 4 13 3 4 13 888 3 012 345 6789 2 12345 67890	3 6 5 6 7 5 16 1230 12391 68273

Explanation for the 1st sample input.

The machine has one command op-code (4) which must not appear in the generated serial numbers; therefore, the sequence of serial numbers will run as follows: 1 2 3 5 6 7 8 9 … The requested items are the 3rd and the 5th item, which respectively have serial numbers 3 and 6.

Problem I

Romantic Date

Wibowo and his girlfriend are playing with a deck of card. A deck consists of 52 different cards. Each card has a number (from lowest to highest: 2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King, Ace) and a suit (from weakest to strongest: Diamond, Club, Heart, Spade). When two cards are compared, the one with higher number is considered win. If they have the same number then the stronger suit will win.

First, Wibowo and his girlfriend split the deck so each one get 26 cards. The game consists of 26 rounds. In each round, both Wibowo and his girlfriend pick one card from their hand at the same time and compare it. The winner get one point for that round.

Given the initial hand, Wibowo wondered what is the highest point he could possibly get by playing that hand with his girlfriend.

Input

The first line of input contains an integer T (T ≤ 100) denoting the number of case. Each case is described in a line which contains 26 cards, the Wibowo's hand. Each card has two characters, the first one is number and the second one is suit. A number can be varied between 2, 3, 4, 5, 6, 7, 8, 9, T (10), J (Jack), Q (Queen), K (King), A (Ace). A suit can be varied between D (Diamond), C (Club), H (Heart), S (Spade). Each card is separated by a single space and all cards are distinct.

Output

For each case, output in a line a single integer the highest point that Wibowo possibly get by playing that hand.

Sample Input

```
3
2D 2C 2H 2S 3D 3C 3H 3S 4D 4C 4H 4S 5D 5C 5H 5S 6D 6C 6H 6S 7D 7C 7H 7S 8D 8C
8H 8S 9D 9C 9H 9S TD TC TH TS JD JC JH JS QD QC QH QS KD KC KH KS AD AC AH AS
2D TC 2C 9S 6H TH TD 8H 6S 3C 5H 3S TS 4C 5S JD 3D 2H 6C 7S 9C 6D 8D 4H 9H 5C
```

Output for Sample Input

```
0
26
11
```

Explanation for the 1st sample input.

Wibowo has all the low cards so he can't possibly get any point at all.

Problem J

Fire Drill

Joko is taking part in a fire drill which is held by the Jakarta Fire Department to recruit new firemen. The drill is about rescuing volunteers (who act as unconscious people) trapped in a building in a limited time. The building has several floors, and the volunteers are scattered throughout the building. Each volunteer has points assigned to her. The fireman candidate should rescue volunteers through carrying them to the exit. The candidate will earn the assigned points for each volunteer he rescued.

Each floor of a building can be seen as a grid of cells. Each cell can be an obstacle, an empty space, a stair or an entry/exit point.

A candidate starts at the entry point which exists only at one single cell of the first floor of the building. The candidate can move to any adjacent non-obstacle cells (north, south, west or east) or climb up or down a stair in 1 second. The movement slows down to 2 seconds when the candidate carries a volunteer. When a candidate finds a volunteer, he may decide to rescue her or not, but if he decides to rescue her, he has to carry her back to the exit without stopping. He can only carry at most one volunteer at a time.

Joko has the floor plan of the test building. Help him plan his moves, so he can get the highest possible score.

Input

The first line of input contains an integer T (T ≤ 100) denoting the number of case. Each case has five integers L (1 ≤ L ≤ 10), H (1 ≤ H ≤ 100), W (1 ≤ W ≤ 100), N (1 ≤ N ≤ 100) and S (1 ≤ S ≤ 10,000) denoting the number of floors, height and weight of each floor, the number of unconscious people, and the given time respectively.

The next L blocks describe the map of each floor from the 1^{st} floor to the L^{th} floor respectively. Each floor consists of H lines each contains W characters. Characters that may appear in each floor are:

- ‘`S`’ : The starting point, also serves as the exit point. There will be only one starting/exit point and it will appear in the first floor.
- ‘`X`’ : Obstacle, cell that cannot be visited (wall, fire, etc.).
- ‘`U`’ : Stair that connect to the upper floor. There will be a ‘`D`’ character at the same place in the upper level. This character will not appear in the highest level of the building.
- ‘`D`’ : Stair that connect to the lower floor. There will be a ‘`U`’ character at the same place in the lower level. This character will not appear in the lowest level of the building.
- ‘`.`’ : Empty space, cell that can be visited.

The next N lines each contains four integers f_i ($1 \le f_i \le L$), r_i ($1 \le ri \le H$), c_i ($1 \le c_i \le W$), p_i ($1 \le p_i \le 1{,}000$) denoting the location of each volunteer (floor, row, column) and the point assigned to this volunteer respectively. You can assume that each volunteer will be located in empty space and no two volunteer occupy the same location.

Output

For each case, output in a line a single integer the highest point that he can earn by rescuing unconscious people within the given time.

Sample Input	Output for Sample Input
2 3 3 5 3 55 XXXXX X..UX XSXXX XXXXX XU.DX XXXXX XXXXX XD..X XXXXX 1 2 3 10 3 2 3 50 3 2 4 60 2 2 6 4 27 ...... S..U.. ....D.. 1 2 3 20 1 2 5 50 1 2 6 50 2 1 1 90	110 100

This page is intentionally left blank