The 43rd ACM International Collegiate Programming Contest
Asia Xuzhou Regional Contest

October 28

Problems

Do not open before the contest has started.

Problem A. Rikka with Minimum Spanning Trees

Input file: `standard input`
Output file: `standard output`

Hello everyone! I am your old friend Rikka. Welcome to Xuzhou. This is the first problem, which is a problem about the minimum spanning tree (MST). I promise you all that this should be the **easiest problem** for most people.

A minimum spanning tree, or minimum weight spanning tree, is a subset of edges from an edge-weighted undirected graph, which forms a tree with the minimum possible total edge weight that connects all the vertices together without any cycles.

In this problem, Rikka wants you to calculate the summation of total edge weights through all MSTs for a given graph, which obviously equals to the product of the total edge weight in an MST and the total number of different MSTs. Note that two spanning trees are different if the sets of their edges are different. In addition, a disconnected graph could have no MSTs, the number of whose different MSTs is zero.

To decrease the size of the input, Rikka provides an edge-weighted undirected graph via a random number generator with given random seeds, denoted by two integers k_1 and k_2. Supposing the number of vertices and edges in the graph are n and m respectively, the following code in C++ tells you how to generate the graph and store the i-th edge between the vertex $u[i]$ and $v[i]$ with weight $w[i]$ in corresponding arrays. You can use the code directly in your submissions.

```
unsigned long long k1, k2;

unsigned long long xorShift128Plus() {
    unsigned long long k3 = k1, k4 = k2;
    k1 = k4;
    k3 ^= k3 << 23;
    k2 = k3 ^ k4 ^ (k3 >> 17) ^ (k4 >> 26);
    return k2 + k4;
}

int n, m, u[100001], v[100001];
unsigned long long w[100001];

void gen() {
    scanf("%d%d%llu%llu", &n, &m, &k1, &k2);
    for(int i = 1; i <= m; ++i) {
        u[i] = xorShift128Plus() % n + 1;
        v[i] = xorShift128Plus() % n + 1;
        w[i] = xorShift128Plus();
    }
}
```

Also, to decrease the size of the output, your code should output the answer modulo $(10^9 + 7)$.

If you have already learned how to handle that, start your show and omit all the rest of the statement.

To make sure everyone knows how to solve this problem, here Rikka would like to provide for you all an effective practice which can solve the problem and help you all get Accepted!

The first one you need to know is the Kirchhoff's matrix tree theorem. Given an undirected graph G with n vertices excluding all loops, its Laplacian matrix $L_{n\times n}$ is defined as $(D - A)$, where D is the degree matrix and A is the adjacency matrix of the graph. More precisely, in the matrix L the entry $l_{i,j}$ $(i \neq j)$

equals to $-m$ where m is the number of edges between the i-th vertex and the j-th vertex, and $L_{i,i}$ equals to the degree of the i-th vertex. Next, construct a matrix L^* by deleting any row and any column from L, for example, deleting row 1 and column 1. The Kirchhoff's matrix tree theorem shows that the number of spanning trees is exactly the determinant of L^*, which can be computed in polynomial time.

Now let me explain an algorithm that counts the number of MSTs. The algorithm breaks up the Kruskal's algorithm for MST into a series of blocks, each of which consists of a sequence of operations about adding edges in a same weight into a multigraph (where a multigraph is a graph, two vertices of which may be connected by more than one edge) whose vertices are components that have been built through the previous block of operations.

Precisely speaking, let's label the multigraph that has been built after the i-th block of operations as G_i. Without loss of generality, let's consider the 0-th block which has no operation and let G_0 be an empty graph with n isolated vertices. The i-th block of operations squeezes vertices in G_{i-1} connected by edges in this block into a single vertex. The result is exactly the graph G_i.

If you know the cardinal principle of Kruskal's algorithm pretty well, you may find that the number of MSTs is the product of the numbers of spanning trees in every component of the graph for each block-defining weight. Actually, the number of edges for a certain weight is fixed in all MSTs, based on the greedy-choice strategy in Kruskal's algorithm. Finally, the Kirchhoff's matrix tree theorem helps you compute the numbers of spanning trees for graphs.

Input

The input contains several test cases, and the first line contains a single integer T ($1 \le T \le 100$), the number of test cases.

For each test case, the only line contains four integers n ($1 \le n \le 10^5$), m ($m = 10^5$), k_1 and k_2 ($10^8 \le k_1, k_2 \le 10^{12}$).

Output

For each test case, output a single line with a single number, the answer modulo $(10^9 + 7)$.

Example

standard input	standard output
1 2 100000 123456789 987654321	575673759

Note

Since the generator code is only provided for C++, Rikka strongly suggests you all solve the problem using C or C++ instead of other programming languages.

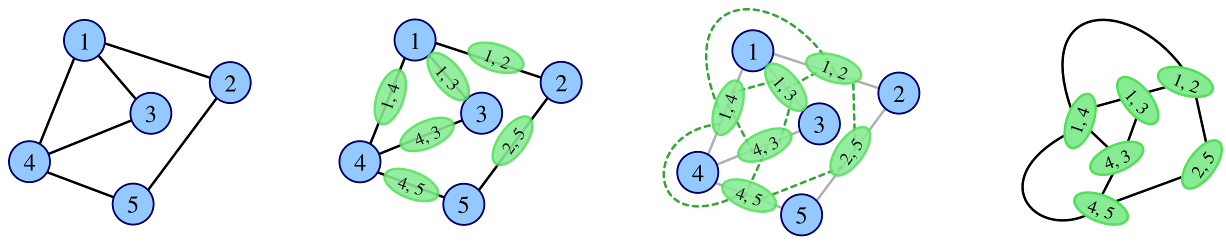
Problem B. Rikka with Line Graphs

Input file:	`standard input`
Output file:	`standard output`

Several years of ACM-ICPC experience enables Rikka, as a student at Keping University, to catch the tide of the algorithm development.

Over the courses for this semester, Rikka made a deep study of line graphs. In the mathematical discipline of graph theory, the line graph of a simple undirected graph G is another simple undirected graph $L(G)$ that represents the adjacency between every two edges in G. Precisely speaking, for an undirected graph G without loops or multiple edges, its line graph $L(G)$ is a graph such that

- each vertex of $L(G)$ represents an edge of G; and
- two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G.



Given a simple undirected graph G, Rikka's study aims to count the number of vertices in its line graph. Now she decides to show you some critical results of her early study, concerning the number of vertices in the line graph of G, $L(G)$, the line graph of the line graph of G, $L^2(G)$ (i. e. $L(L(G))$), and so forth, denoted by $|V(L(G))|$, $|V(L^2(G))|$, $\cdots$.

By the definition of an undirected graph with n vertices and m edges, we know that

$$|V(L(G))| = \sum_e 1 = m = \frac{1}{2}\sum_u d_1(u),$$

where $d_1(u)$ represents the degree of vertex u in G.

Once we know how to count, for any edge e in G, the number of edges which share a common endpoint with e, or equally speaking the degree of e in $L(G)$, which is denoted by $d_1'(e)$, we have

$$|V(L^2(G))| = \frac{1}{2}\sum_e d_1'(e) = \frac{1}{2}\sum_{e=(u,v)}(d_1(u) - 1 + d_1(v) - 1) = \frac{1}{2}\sum_u d_1(u)(d_1(u) - 1).$$

A similar easy analysis can help us to calculate $|V(L^3(G))|$, and an excellent result in Rikka's known work, which was published in the 2018 JheZiang Olympiad in Informatics, reveals the number of vertices in $L^4(G)$ as

$$\begin{aligned}|V(L^4(G))| = \frac{1}{2}\sum_u &(2d_1^2(u) - 13d_1(u) + 21 + 4d_2(u))d_1(u)(d_1(u) - 1)\\ &- 13(d_1(u) - 1)d_2(u) + (d_1(u) - 2)d_{2,2}(u) + d_2^2(u),\end{aligned}$$

where $d_2(u)$ is the summation of degrees of all adjacent vertices of u in G, and when considering the degrees squared of all adjacent vertices of u, $d_{2,2}(u)$ is the summation of them all.

Based on the equation $L^5(G) = L^4(L(G))$, her newest work made a further development. She extrapolates from the result about $L^4(G)$ a linear computable method to calculate the number of vertices in $L^5(G)$ in time complexity $O(n+m)$. Rikka pointed out that the data about vertices required in the summation form of $|V(L^4(G))|$ imply new data about edges with similar definitions. Actually, the relationship between d_1 and d'_1 is the easiest correspondence. A harder one is described as the one between d_2 and d'_2. Luckily, we can calculate all these new data which we need about the edges in linear time. Thus an attempt replacing the summation of vertices by a summation of edges provides a strict formula for the number of vertices in $L^5(G)$.

Now you must try to go with the current of the times. In this problem, for an undirected simple graph G, you are asked to calculate the number of vertices in $L^6(G)$ and output the number modulo $(10^9 + 7)$.

Input

The input contains several test cases, and the first line contains a single integer T $(1 \le T \le 10)$, the number of test cases.

For each test case, the first line contains two integers n $(1 \le n \le 10^5)$ and m $(0 \le m \le 2 \times 10^5)$, the number of vertices and edges in the given simple undirected graph G.

Then m lines follow, describing all edges of the graph. Each line of them contains two integers u and v $(1 \le u, v \le n,\ u \ne v)$, representing an edge between the u-th vertex and the v-th vertex.

The input guarantees that the given graph for each test case contains no loops or multiple edges.

Output

For each test case, output a single line with a single integer, the remainder of the number of vertices in $L^6(G)$ divided by $(10^9 + 7)$.

Example

standard input	standard output
2 4 4 1 2 2 3 3 1 4 1 4 4 1 2 2 3 3 4 4 1	396 4

Problem C. Rikka with Consistency

Input file:	`standard input`
Output file:	`standard output`

On the way to the Moscow, Rikka knows someone will replace her. Who is the guy? A devil to get in touch with her dark side, or an angel to rinse the shadow off her mind? However, Rikka knows that her successor has a special name, whose meaning in Chinese is Consistency. The process of replacement is so wonderful and sentimental, which is what you all must know.

Now, the only road from Beijing to Moscow is described as a broken line with n segments in the X-H plane. The i-th segment connects the points $(i-1, h_{i-1})$ and (i, h_i), and $h_0 = h_n = 0$ are known. This figure is a topographic map showing the whole trip from Beijing to Moscow and its H axis indicates the altitude. The distance of a path between two points is the length of the broken line between their corresponding points in the map.

At the outset of the trip, Rikka is in Beijing whose location in the X-H plane is $(0, 0)$; Consistency, the guy who will replace Rikka, is in Moscow which is located at $(n, 0)$. Consistency always maintains consistent academic standards, a consistent life level, a consistent height of perspective and the altitude as what Rikka owns. This is why their heights are the same yesterday, today and forever.

Now Rikka wants you to calculate the minimum total distance they need (which is the total length of paths that Rikka and Consistency travel along). By the time that Rikka arrives in Moscow and Consistency arrives in Beijing as well, their replacement will be finished (and this is the ending which is also a new beginning).

Input

The input contains several test cases, and the first line contains a single integer T $(1 \le T \le 500)$, the number of test cases.

For each test case, the first line contains a single integer n $(1 \le n \le 50)$, the number of segments.

The second line contains $(n+1)$ integers $h_0, h_1, \cdots, h_n$ $(0 \le h_i \le 50)$, satisfying $h_0 = h_n = 0$.

The input guarantees that the paths for each test case always exist.

Output

For each test case, output a single line with a single number, the minimum total distance they need. Your answer is considered correct if its absolute or relative error does not exceed 10^{-9}. Formally, let your answer be a, and Rikka's answer be b. Your answer is considered correct if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Example

standard input	standard output
2 4 0 1 1 2 0 4 0 2 1 3 0	12.128990204491960 22.313624568639947

This page is intentionally left blank

Problem D. Rikka with Subsequences

Input file: `standard input`
Output file: `standard output`

For a known sequence, counting subsequences with a certain remarkable property can depict the sequence itself to a certain extent.

Now, Rikka has a sequence A of length n whose elements, denoted by $a_1, a_2, \cdots, a_n$, are positive integers in $[1, n]$. A bridging relation matrix (BRM) is a $n \times n$ logical matrix with elements from $\{0, 1\}$. Here Rikka defines a Yuta subsequence based on a given BRM, $M = (M_{i,j})_{1 \le i,j \le n}$.

Rikka calls a subsequence of A, denoted by $a_{p_1}, a_{p_2}, \cdots, a_{p_m}$ with $1 \le p_1 < p_2 < \cdots < p_m \le n$, a Yuta subsequence if and only if $M_{a_{p_i}, a_{p_{i+1}}} = 1$ for $i = 1, 2, \cdots, m-1$. Counting the number of different Yuta subsequences has a profound value in data analysis and data recovery.

Rikka thinks this task is too simple and she wants to make it look harder and more heuristic. Rikka knows that a Yuta subsequence may appear in the sequence A several times and top programmers may use something like `map<vector<int>, bigInt> cnt` in C++ or `Map<ArrayList<Integer>, BigInteger> cnt` in Java to store all Yuta subsequences and count the numbers.

She calls the sum of the cubes of the numbers of occurrences for all Yuta subsequences, which is equal to the sum of cubes of all the second elements in `cnt`, the third coefficient of A over M.

Now, after showing you the sequence and the BRM, she wants you to calculate the third coefficient of the sequence over the given BRM in modulo $(10^9 + 7)$.

Input

The input contains several test cases, and the first line contains a single integer T $(1 \le T \le 20)$, the number of test cases.

For each test case, the first line contains a single integer n $(1 \le n \le 200)$, the length of the sequence A.

The second line contains n integers $a_1, a_2, \cdots, a_n$ $(1 \le a_i \le n)$.

The following n lines describe the given BRM, where each line of them has n characters, and the j-th character in the i-th line of them is either 0 or 1, representing the element $M_{i,j}$.

Output

For each test case, output a single line with a single integer, the third coefficient of the given sequence over the given BRM in modulo $(10^9 + 7)$.

Example

standard input	standard output
1 4 1 2 1 2 1111 1111 1111 1111	51

This page is intentionally left blank

Problem E. Rikka with Data Structures

Input file: `standard input`
Output file: `standard output`

As we know, Rikka is poor at data structures. Yuta is worrying about this situation, so he gives Rikka some tasks about data structures to practice. Here is one of them:

Yuta has an array A with n numbers, denoted by $A[1], A[2], \cdots, A[n]$. Then he makes m operations on it.

There are three types of operations:

- `1 l r k`: for each index i in $[l, r]$, change the value of $A[i]$ into $(A[i] + k)$;
- `2 l r k`: for each index i in $[l, r]$, change the value of $A[i]$ into k;
- `3 l r x`: Yuta wants Rikka to count the number of different indices y with $l \leq y \leq r$ such that $\max\{A[\min\{x, y\}], A[\min\{x, y\} + 1], \cdots, A[\max\{x, y\}]\} = \max\{A[x], A[y]\}$.

It is too difficult for Rikka. Can you help her?

Input

The input contains several test cases, and the first line contains a single integer T $(1 \leq T \leq 200)$, the number of test cases.

For each test case, the first line contains two integers n $(1 \leq n \leq 10^5)$ and m $(1 \leq m \leq 10^5)$.

The second line contains n integers $A[1], A[2], \cdots, A[n]$ $(1 \leq A[i] \leq 10^9)$.

Then m lines follow, each line of which describes an operation, containing four integers as mentioned above, satisfying $1 \leq l \leq r \leq n$, $1 \leq k \leq 10^9$ and $1 \leq x \leq n$.

The input guarantees that there are at most 10 test cases with $n > 10^3$ or $m > 10^3$.

Output

For each query, an operation of type 3, output a single line with a single integer, the answer to this query.

Example

standard input	standard output
1 10 10 1 3 2 5 2 3 1 6 4 5 3 5 7 8 3 5 7 4 1 1 5 2 3 1 10 4 3 1 10 8 2 8 8 8 3 1 10 8 3 1 10 4 2 4 8 1 3 1 2 10	3 3 10 7 10 8 2

This page is intentionally left blank

Problem F. Rikka with Nice Counting Striking Back

Input file: `standard input`
Output file: `standard output`

As we know, Yuta is poor at counting numbers. Rikka is worrying about this situation, so she gives Yuta some counting tasks to practice. Here is one of them:

In computer programming, a string is traditionally a sequence of characters and a substring of a string is a contiguous sequence of characters within the string. For instance, `snowball` is a string, `now` is a substring of `snowball` and `bow` is not a substring of `snowball`. Moreover, the concatenation of two strings U and V is named as UV, that is, if U is `snow` and V is `ball`, then UV is `snowball`.

Rikka has a string S of length n and she wants Yuta to count how many distinct *nice* strings in total. Here, she calls a non-empty string T *nice* if

- T is a substring of S; and
- TP is not a substring of S for any non-empty string P meeting the condition that TP and PT are the same string.

It is too difficult for Yuta. Can you help him?

Input

The input contains several test cases, and the first line contains a single integer T ($1 \le T \le 1000$), the number of test cases.

For each test case, the only line contains a single string S of length n ($1 \le n \le 2 \times 10^5$) with only lowercase letters.

The input guarantees that the sum of n in all test cases is at most 5×10^6.

Output

For each test case, output a single line with a single integer, the answer.

Example

standard input	standard output
6 rikkasuggeststoallthecontestants thisisaproblemdesignedforgrandmasters ifyoudidnotachievethat youdbetterskiptheproblem wishyouahighrank enjoytheexperience	500 679 244 290 132 163

This page is intentionally left blank

Problem G. Rikka with Intersections of Paths

Input file: `standard input`
Output file: `standard output`

Rikka has a tree T with n vertices numbered from 1 to n.

Meanwhile, Rikka has marked m simple paths in T, the i-th of which is between the vertices x_i and y_i, where some of them could be the same path.

Now, Rikka wants to know in how many different strategies she can select k paths from the marked paths such that those selected paths share at least one common vertex.

Input

The input contains several test cases, and the first line contains a single integer T ($1 \le T \le 200$), the number of test cases.

For each test case, the first line contains three integers n ($1 \le n \le 3 \times 10^5$), the size of the tree T, m ($2 \le m \le 3 \times 10^5$), the number of marked paths, and k ($2 \le k \le m$).

The following $(n-1)$ lines describe the tree T. Each of them contains two integers u and v ($1 \le u, v \le n$, $u \ne v$), representing an edge between the vertices u and v.

The following m lines describe all marked simple paths in the tree. The i-th of them contains two integers x_i and y_i ($1 \le x_i, y_i \le n$).

The input guarantees that the sum of n and the sum of m in all test cases are at most 2×10^6 respectively.

Output

For each test case, output a single line with a single integer, the number of different strategies meeting the requirement modulo ($10^9 + 7$).

Example

standard input	standard output
1 3 6 2 1 2 1 3 1 1 2 2 3 3 1 2 1 3 2 3	10

This page is intentionally left blank

Problem H. Rikka with A Long Colour Palette

Input file: `standard input`
Output file: `standard output`

Blue, the colour of the sky, the sea and your eyes.

Green, the colour of nature, fertility and life.

Purple, the colour of good judgment, and of people seeking spiritual fulfillment.

Orange, the only colour that is also a fruit.

Yellow, the colour in smiley faces.

Red, the warmest of all.

Rikka loves them all, but what is her favourite colour? She has found k different colours, numbered from 1 to k, and she knows that the best colour should be the one after fixing them all together. The best colour is called the DREAM.

Rikka has also prepared a long and narrow colour palette of length 10^9. She designates n segments in the palette. A segment described by two integers l and r ($0 \leq l < r \leq 10^9$) represents an area of the palette where the distance between the leftmost end of the palette and the left endpoint (resp. the right endpoint) of the area is l (resp. r).

She will for each segment designated smear the pigment of any of these colours (from 1 to k) which she has found evenly on it. Some areas may contain pigments of several different colours, since these segments may intersect. If some areas contain all these k different colours which she has found, it would blend into the DREAM.

Now, Rikka wants you to maximize the total length of all areas in the palette such that each part of them can blend into the DREAM. You also need to provide a feasible plan.

Input

The input contains several test cases, and the first line contains a single integer T ($1 \leq T \leq 1000$), the number of test cases.

For each test case, the first line contains two integers n ($1 \leq n \leq 2 \times 10^5$), the number of segments designated by Rikka, and k ($1 \leq k \leq 2 \times 10^5$), the number of colours which Rikka has found.

Each of the following n lines contains two integers l and r ($0 \leq l < r \leq 10^9$), representing the i-th segment in the palette.

The input guarantees that the sum of n in all test cases is at most 2×10^6.

Output

For each test case, output two lines. Firstly, output a line with a single integer, the largest total length of areas required. Then, output a line with n space-separated integers describing a feasible plan, where the i-th number is the colour for the i-th segment.

All feasible plans are allowed, so you can output any of them.

Example

standard input	standard output
1 3 2 1 5 2 4 3 6	3 1 2 2

This page is intentionally left blank

Problem I. Rikka with Sorting Networks

Input file: `standard input`
Output file: `standard output`

Rikka knows that Bubble sort is a simple but beautiful algorithm, Quicksort is a complex but efficient algorithm, and Shellsort is a weird but practical algorithm. Rikka is interested in all sorting algorithms and she can assign as many new problems for ICPC contests as she wants.

Rikka hates those guys who create new problems with the same ideas over and over again, and she hopes not to become the person she hates to be. Though she has already assigned several problems for sorting algorithms such as Merge sort and Insertion sort, she decides to show you the last problem about sorting algorithms to end this series forever.

Here Rikka introduces the sorting network and she defines a comparator at first. For a permutation A of the n smallest positive integers denoted by $a_1, a_2, \cdots, a_n$, a comparator $[u, v]$ $(u \neq v)$ sorts the u-th and the v-th element in A into nondecreasing order. Formally, a comparator is a mapping $[u, v]$ satisfying

- $[u, v](a_u) = \min(a_u, a_v)$; and
- $[u, v](a_v) = \max(a_u, a_v)$; and
- $[u, v](a_k) = a_k$ for all k with $k \neq u$ and $k \neq v$.

Rikka defines a sorting network as a composition of comparators and provides for you a sorting network with k comparators. Now, Rikka wants you to count the number of permutations of 1 to n which, through the given sorting network, would become an almost sorted permutation. She says a permutation of 1 to n is almost sorted if the length of its longest increasing subsequence is at least $(n - 1)$.

Input

The input contains several test cases, and the first line contains a single integer T $(1 \leq T \leq 100)$, the number of test cases.

For each test case, the first line contains three integers n $(2 \leq n \leq 50)$, the length of permutations, k $(0 \leq k \leq 10)$, the number of comparators, and q $(10^8 \leq q \leq 10^9)$, a prime number for the output.

Then k lines follow, the i-th line of which contains two integers u and v $(1 \leq u < v \leq n)$, representing the i-th comparator $[u, v]$.

Output

For each test case, output a single line with a single integer, the remainder of the number of permutations which meet the requirement divided by q.

Example

standard input	standard output
4 4 0 998244353 4 1 998244353 1 2 4 3 998244353 1 2 2 3 1 2 4 6 998244353 1 2 2 3 1 2 3 4 2 3 1 2	10 14 24 24

Problem J. Rikka with An Unnamed Temple

Input file: standard input
Output file: standard output

Rikka discovered an unnamed ancient temple together with its internal map by accident. The temple contains n separated rooms. Several one-way roads connect these rooms and form a directed acyclic graph.

When a visitor enters the temple, she will appear in the first room. she can find the exit of the temple in the n-th room. Notice that she probably cannot arrive all rooms from the entrance and meanwhile, she may not have the chance to escape the temple from some room inside.

All rooms have some treasures. The weight of the treasure stored in the i-th room is w_i, and its value is c_i. A visitor, when she arrives at the exit, is allowed to leave the temple if the remainder of the total weight of treasures she has picked divided by k is equal to t where k and t are fixed integers.

Besides, a guardian is standing in a room and protecting the treasure, but no one knows where she is. To prevent being attacked, visitors should not step into the room of the guardian at any time.

Now Rikka decides to visit the unnamed temple. She will select a path from the entrance to the exit, picking measures in all rooms she will pass through. She wants you to calculate, for each index i from 1 to n, what the maximum total value she can obtain is and in how many ways she could achieve that, in case the guardian is standing in the i-th room.

Input

The input contains several test cases, and the first line contains a single integer T ($1 \leq T \leq 1000$), the number of test cases.

For each test case, the first line contains two integers n ($2 \leq n \leq 10^5$), the number of rooms, and m ($0 \leq m \leq 2 \times 10^5$), the number of one-way roads.

The following n lines describe all rooms. The i-th of them contains two integers w_i and c_i ($1 \leq w_i, c_i \leq 10^9$).

Then following n lines describe all roads. The i-th of them contains two integers u and v ($1 \leq u < v \leq n$) which describes a one-way road from the u-th room to the v-th room.

The last line contains two integers k and t ($0 \leq t < k \leq 100$) which are the coefficients for the condition to leave the temple.

The input guarantees that all roads in a single test case are distinct, the sum of n in all test cases is at most 10^6, and the sum of m in all test cases is at most 2×10^6.

Output

For each test case, output n lines. In the i-th line, we consider the case when the guardian is standing in the i-th room. If there is no valid path for Rikka to visit the temple from the entrance to the exit, output -1 in this line. Otherwise, output two space-separated integers in this line, where the first one is the maximum total value she can obtain, and the second one is the number of different paths she can select to achieve the best result. The first number should be outputted in exact form, while the second one should be outputted in modulo $(10^9 + 7)$.

Example

standard input	standard output
1 4 5 1 2 2 3 3 4 4 2 1 2 1 3 2 4 3 4 1 4 5 3	-1 8 1 -1 -1

Problem K. Rikka with Ants

Input file: standard input
Output file: standard output

Every time when Rikka faces to a great nature sight, she will recall a line of ants moving in a hurry. Rikka loves ants and keeps two large colonies of ants in her drawer. Observing ants hurrying in moving has controlled her life.

That is why she prepares n different nests in her drawer, and n undirected channels between these nests form a circular orbit. All nests are numbered by 1 to n in order, and the lengths of channels are known. The first colony of ants is living in the s_1-th nest, and the second colony is living in the s_2-th nest.

Now, ants decide to move to new nests together. The new home for these two colonies of ants will be the e_1-th nest and the e_2-th nest respectively.

For each colony, all ants should queue up one by one to crawl from the origin to the destination along a path. They cannot be split into several groups crawling moving along different paths. Then, they can measure the complexity of their plan moving to new nest using the total length of channels in the path selected.

If these two colonies of ants select paths sharing some common channels, they will walk through these channels slowly for security. Specifically, for each common channel, we can consider equivalently that its measured length will be tripled.

Ants are highly intelligent and they all want to minimize the complexities of their plan. They will choose the best strategies for themselves respectively without negotiation. All they know are the lengths of channels, and the nests where their colony and the other one will start and end respectively.

Rikka wants you to calculate the expected complexities of plans for each colony of ants.

Input

The input contains several test cases, and the first line contains a single integer T ($1 \le T \le 5000$), the number of test cases.

For each test case, the first line contains a single integer n ($2 \le n \le 50$), the number of nests and also the number of channels between them.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_i \le 50$), where the i-th one, a_i, represents the length of the undirected channel between the i-th nest and the $((i \bmod n) + 1)$-th nest.

The third line contains four integers s_1, e_1, s_2 and e_2 ($1 \le s_1, s_2, e_1, e_2 \le n$, $s_1 \ne e_1$, $s_2 \ne e_2$), representing the original nests where these ants live and their new nests.

Output

For each test case, output a line with two space-separated numbers, representing the expected complexity for the first colony of ant and the expected complexity for the second colony respectively. Your answer is considered correct if the absolute or relative error between each number in your output and the corresponding one in Rikka's answer does not exceed 10^{-9}. Formally, let a number of your answer be a, and the corresponding number of Rikka's answer be b. Your answer is considered correct if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Example

standard input	standard output
2 5 1 5 2 4 3 1 2 3 4 5 1 5 2 4 3 1 3 2 4	1.000000000000000 2.000000000000000 14.666666666666667 14.666666666666667

Note

What we are talking about including strategies, best strategies and expectations are actually what about Nash Equilibrium and mixed strategies.

In the theory of games, a player is said to use a mixed strategy whenever the player chooses to randomize over the set of available actions. Formally, a mixed strategy is a probability distribution that assigns to each available action a likelihood of being selected. If only one action has a positive probability of being selected, the player is said to use a pure strategy. In the first sample case, the best strategies for both colonies are pure strategies.

A mixed strategy profile is a list of strategies, one for each player in the game. A mixed strategy profile induces a probability distribution or lottery over the possible outcomes of the game. In this problem, the profiles are those plans that we discussed before.

A Nash equilibrium (mixed strategy) is a strategy profile with the property that no single player can, by deviating unilaterally to another strategy, induce a lottery that he or she finds strictly preferable. In 1950, the mathematician John Nash proved that every game with a finite set of players and actions has at least one equilibrium.

In this problem, a Nash equilibrium is what you need to find.

Problem L. Rikka with Grid Graphs

Input file: `standard input`
Output file: `standard output`

A two-dimensional grid graph, also known as a square grid graph, is an undirected graph with nm vertices corresponding all nodes in a $n \times m$ grid. An edge here is an abstract description for a matchstick of length one in the grid connecting two adjacent nodes.

Now, Rikka gives you a $n \times m$ grid graph made with no more than $(2nm - n - m)$ edges. She wants you to calculate the number of different orientations for the undirected graph such that the oriented graph is a directed graph with no directed cycles.

In this problem, an orientation of an undirected graph is an assignment of a direction to each edge, turning the initial graph into a directed graph.

Input

The input contains several test cases, and the first line contains a single integer T $(1 \le T \le 60)$, the number of test cases.

For each test case, the first line contains two integers n and m $(1 \le n, m \le 6)$, the number of vertices in one column and one row of the grid graph respectively.

Then $(2n - 1)$ lines follow, each of which has at most $(2m - 1)$ characters, describing the grid graph. The odd lines of them contain grid vertices, represented by separated plus signs (`+`), and zero or more horizontal edges, while the even lines of them contain zero or more vertical edges. Specifically, all possible vertices are represented in the input. Each horizontal edge connecting neighbouring vertices is represented by a single minus sign (`-`), while each vertical edge is represented by a single vertical bar (`|`). The edge characters will be placed exactly between the corresponding vertices. All other characters will be space characters.

Note that if any input line could contain trailing whitespace, that whitespace may be omitted.

Output

For each test case, output a single line with a single integer, the number of valid orientations for the given grid graph.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>4
2 2
+-+

+ +
2 2
+-+
|
+ +
2 2
+-+
|
+-+
2 2
+-+
| |
+-+</pre></td><td><pre>2
4
8
14</pre></td></tr>
</table>

This page is intentionally left blank

Problem M. Rikka with Illuminations

Input file: `standard input`
Output file: `standard output`

Rikka loves convex polygons, so she decides to install some illuminants to embellish polygons.

Now she has a large convex polygon with n sides. She also has m different points strictly outside the polygon which are all legal positions to install illuminants.

An illuminant can light up some exterior boundaries of the polygon.

Rikka wants to install some illuminants to light up all exterior boundaries of the polygon. She asks you to calculate the least number of illuminants which she needs and provide a feasible scheme.

Input

The input contains several test cases, and the first line contains a single integer T ($1 \le T \le 100$), the number of test cases.

For each test case, the first line contains two integers n ($3 \le n \le 1000$) and m ($1 \le m \le 1000$).

Each of the following n lines describes a vertex on the convex polygon with two integers x and y ($|x|, |y| \le 10^9$), the Cartesian coordinates of the vertex. All these vertices are given in counter-clockwise order and any three of them are not collinear.

Then the following m lines contain m different points outside the polygon describing all legal positions to install illuminants. Each of them contains two integers x and y ($|x|, |y| \le 10^9$), the Cartesian coordinates of a legal position. They are numbered from 1 to m. All these positions would not lie in some extension lines for the sides of the polygon.

Output

For each test case, if it's impossible to light up all exterior boundaries of the polygon, output a single line with a single integer -1. Otherwise, output two lines. Firstly, output a line with a single integer k, representing the least number of illuminants Rikka needs to light up all the boundaries. Then, output a line with k space-separated distinct integers, describing a feasible scheme, each of which is the index of a selected position.

All feasible schemes are allowed, so you can output any of them.

Example

standard input	standard output
1 3 3 0 0 1 0 0 1 -1 -1 3 -1 -1 3	2 1 2

This page is intentionally left blank