Problem A. Acperience

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Deep neural networks (DNN) have shown significant improvements in several application domains including computer vision and speech recognition. In computer vision, a particular type of DNN, known as Convolutional Neural Networks (CNN), have demonstrated state-of-the-art results in object recognition and detection.

Convolutional neural networks show reliable results on object recognition and detection that are useful in real world applications. Concurrent to the recent progress in recognition, interesting advancements have been happening in virtual reality (VR by Oculus), augmented reality (AR by HoloLens), and smart wearable devices. Putting these two pieces together, we argue that it is the right time to equip smart portable devices with the power of state-of-the-art recognition systems. However, CNN-based recognition systems need large amounts of memory and computational power. While they perform well on expensive, GPU-based machines, they are often unsuitable for smaller devices like cell phones and embedded electronics.

In order to simplify the networks, Professor Zhang tries to introduce simple, efficient, and accurate approximations to CNNs by binarizing the weights. Professor Zhang needs your help.

More specifically, you are given a weight vector $W = (w_1, w_2, \ldots, w_n)$. Professor Zhang would like to find a binary vector $B = (b_1, b_2, \ldots, b_n)$ $(b_i \in \{-1, +1\})$ and a real scaling factor $\alpha \geq 0$ in such a manner that $\|W - \alpha B\|^2$ will be minimum possible.

Note that $\|\cdot\|$ denotes the Euclidean norm, that is, $\|X\| = \sqrt{x_1^2 + \cdots + x_n^2}$, where $X = (x_1, x_2, \ldots, x_n)$.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n $(1 \leq n \leq 100\,000)$: the length of the given weight vector. The next line contains n integers: $w_1, w_2, \ldots, w_n$ $(-10\,000 \leq w_i \leq 10\,000)$.

There are no more than 400 test cases. The total size of the input is at most 7 mebibytes.

Output

For each test case, output the minimum value of $\|W - \alpha B\|^2$ as an irreducible fraction p/q where p and q are integers, and $q > 0$.

Example

standard input	standard output
3 4 1 2 3 4 4 2 2 2 2 5 5 6 2 3 4	5/1 0/1 10/1

Problem B. Born Slippy

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7.5 seconds
Memory limit:	256 mebibytes

Professor Zhang has a rooted tree with vertices conveniently labeled by $1, 2, \ldots, n$. The i-th vertex has an integer weight w_i.

For each $s \in \{1, 2, \ldots, n\}$, Professor Zhang wants to find a sequence of vertices $v_1, v_2, \ldots, v_m$ such that:

- $v_1 = s$ and v_i is the ancestor of v_{i-1} for each $1 < i \le m$,
- the value $f(s) = w_{v_1} + \sum\limits_{i=2}^{m} (w_{v_i} \text{ op } w_{v_{i-1}})$ is maximum possible. Here, operation x op y is the bitwise AND, OR, or XOR operation on two integers.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n and a string op ($2 \le n \le 2^{16}$, op $\in \{\texttt{AND}, \texttt{OR}, \texttt{XOR}\}$): the number of vertices and the operation. The second line contains n integers $w_1, w_2, \ldots, w_n$ ($0 \le w_i < 2^{16}$). The third line contains $n-1$ integers $p_2, p_3, \ldots, p_n$ ($1 \le p_i < i$) where p_i is the parent of vertex i.

There are about 300 test cases, and the sum of n in all the test cases is no more than 10^6.

Output

For each test case, output the integer $S = (\sum\limits_{i=1}^{n} i \cdot f(i))$ modulo $10^9 + 7$.

Example

standard input	standard output
3 5 AND 5 4 3 2 1 1 2 2 4 5 XOR 5 4 3 2 1 1 2 2 4 5 OR 5 4 3 2 1 1 2 2 4	91 139 195

Problem C. Call It What You Want

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Professor Zhang has heard that the longest path problem cannot be solved in polynomial time for arbitrary graphs unless P = NP. Now, Professor Zhang would like to solve this problem in polynomial time in some graphs.

The longest path problem is the problem of finding a simple path of maximum length in a given graph. A path is called simple if it does not have any repeated vertices. The length of a path is the number of edges in this path.

Input

There are multiple test cases. The first line of input contains an integer T (about 350) indicating the number of test cases. For each test case:

The first line contains two integers n and m ($3 \le n \le 10^4$, $n \le m \le n + 4$): the number of vertices and the number of edges.

Each of the following m lines contains two integers a_i and b_i which denotes an edge between vertices a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$).

It is guaranteed that the graph is connected and does not contain multiple edges.

The total size of the input is at most 4 mebibytes.

Output

For each test case, output an integer denoting the length of the longest path.

Example

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

Problem D. Differencia

Input file:	*standard input*
Output file:	*standard output*
Time limit:	30 seconds
Memory limit:	256 mebibytes

Professor Zhang has two sequences $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$. He wants to perform two kinds of operations on the sequences:

- `+` l r x: set a_i to x for all $l \le i \le r$.
- `?` l r: find the number of i such that $a_i \ge b_i$ and $l \le i \le r$.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains four integers n, m, A and B ($1 \le n \le 10^5$, $1 \le m \le 3\,000\,000$, $1 \le A, B \le 2^{16}$): the length of the sequence, the number of operations and two parameters. The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$). The third line contains n integers $b_1, b_2, \ldots, b_n$ ($1 \le b_i \le 10^9$).

As the number of operations can be rather large, the m operations are specified by parameters A and B given to the following generator routine.

```
int a = A, b = B, C = ~(1<<31), M = (1<<16)-1;
int rnd(int last) {
  a = (36969 + (last >> 3)) * (a & M) + (a >> 16);
  b = (18000 + (last >> 3)) * (b & M) + (b >> 16);
  return (C & ((a << 16) + b)) % 1000000000;
}
```

For the i-th operation, first call `rnd`($last$) three times to get l, r and x (that is, $l = \texttt{rnd}(last) \bmod n + 1$, $r = \texttt{rnd}(last) \bmod n + 1$, $x = \texttt{rnd}(last) + 1$). Then, if $l > r$, you should swap their values. And at last, the i-th operation has type '`?`' if $(l + r + x)$ is an even number, or type '`+`' otherwise.

Note: $last$ is the answer of the latest type '`?`' operation. Assume $last = 0$ at the beginning of each test case.

There are at most 300 test cases, and the total size of the input is at most 8 mebibytes.

Output

For each test case, output the integer $S = (\sum_{i=1}^{m} i \cdot z_i) \bmod (10^9 + 7)$, where z_i is the answer for i-th query. If the i-th query is of type '`+`', assume $z_i = 0$.

Example

standard input	standard output
3 5 10 1 2 5 4 3 2 1 1 2 3 4 5 5 10 3 4 5 4 4 2 1 1 2 3 4 5 5 10 5 6 5 4 5 2 1 1 2 2 4 5	81 88 87

Problem E. Eureka

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	64 mebibytes

Professor Zhang draws n points on the plane which are conveniently labeled by $1, 2, \ldots, n$. The i-th point is at (x_i, y_i). Professor Zhang wants to know the number of *best sets*. As the value could be very large, print it modulo $10^9 + 7$.

A set P (P contains the labels of the points) is called a *best set* if and only if there is at least one *best pair* in P. Two numbers u and v ($u, v \in P, u \neq v$) are called a *best pair* if for every $w \in P$, $f(u, v) \geq g(u, v, w)$, where $f(u, v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$ and $g(u, v, w) = \frac{f(u,v)+f(v,w)+f(w,u)}{2}$.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \leq n \leq 1000$): the number of points.

Each of the following n lines contains two integers x_i and y_i ($-10^9 \leq x_i, y_i \leq 10^9$): coordinates of the i-th point.

There are no more than 250 test cases, and the sum of n in all the test cases is at most 40 000.

Output

For each test case, output a single integer: the number of *best sets* modulo $10^9 + 7$.

Example

standard input	standard output
3 3 1 1 1 1 1 1 3 0 0 0 1 1 0 1 0 0	4 3 0

Problem F. Fantasia

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7.5 seconds
Memory limit:	64 mebibytes

Professor Zhang has an undirected graph G with n vertices and m edges. Each vertex has an integer weight w_i. Let G_i be the graph obtained by deleting the i-th vertex from graph G. Professor Zhang wants to find the weights of $G_1, G_2, \ldots, G_n$.

The weight of a graph G is defined as follows:

- If G is connected, then the weight of G is the product of the weight of each vertex in G.
- Otherwise, the weight of G is the sum of the weights of all the connected components of G.

A connected component H of an undirected graph G is a subgraph in which any two vertices are connected by a path, and no other vertex in G is connected to any vertex from H by a path.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($2 \le n \le 10^5$, $1 \le m \le 2 \cdot 10^5$): the number of vertices and the number of edges.

The second line contains n integers $w_1, w_2, \ldots, w_n$ ($1 \le w_i \le 10^9$) denoting the weight of each vertex.

Each of the next m lines contains two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$) denoting an undirected edge.

There are at most 1000 test cases, the sum of n in all the test cases is at most $1.5 \cdot 10^6$, and the sum of m in all the test cases is also at most $1.5 \cdot 10^6$.

Output

For each test case, output the integer $S = (\sum_{i=1}^{n} i \cdot z_i)$ modulo $10^9 + 7$, where z_i is the weight of G_i.

Example

standard input	standard output
1 3 2 1 2 3 1 2 2 3	20

Problem G. Glorious Brilliance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Professor Zhang is trying to solve one of Karp's 21 NP-complete problems: the *Graph Coloring Problem*.

At first, he generates an undirected graph with n vertices and m edges. Then, he colors all the vertices black or white. Finally, he wants to use the following operation to make the vertices correctly colored: choose two adjacent vertices and swap their colors. The vertices are correctly colored if and only if no two adjacent vertices share the same color.

Professor Zhang wants to know the minimum number of operations needed.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($2 \le n \le 500$, $1 \le m \le \frac{n \cdot (n-1)}{2}$): the number of vertices and the number of edges. The second line contains a binary string of length n. The i-th vertex is colored white if the i-th character is '0', or black otherwise.

Each of the next m lines contains two integers x_i and y_i ($1 \le x_i, y_i \le n$, $x_i \ne y_i$) denoting an undirected edge.

There are at most 200 test cases, and the total size of the input is no more than 1.5 mebibytes.

Output

For each test case, output an integer s denoting the minimum number of operations in the first line. Each of the next s lines must contain two integers u_i and v_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$) denoting the i-th operation.

If there is no such solution, just output "-1" on a single line.

Example

standard input	standard output
3 4 4 0011 1 2 2 3 3 4 4 1 2 1 00 1 2 6 7 011001 1 4 1 5 4 2 5 2 5 3 2 6 6 3	1 4 1 -1 2 2 4 3 5

Problem H. Helter Skelter

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	128 mebibytes

A non-empty string s is called a *binary string* if it consists only of characters '`0`' and '`1`'. A substring $s[l \dots r]$ $(1 \leq l \leq r \leq |s|)$ of string $s = s_1 s_2 \dots s_{|s|}$ (where $|s|$ is the length of string s) is the string $s_l s_{l+1} \dots s_r$.

Professor Zhang has got a long binary string s starting with '`0`', and he wants to know whether there is a substring of s such that the number of occurrences of '`0`' and '`1`' in this substring are exactly a and b, respectively, where a and b are two given integers.

Since the binary string is very long, we will compress it. The compression method is as follows:

- Split the string into runs of equal consecutive characters.
- Any two adjacent runs consist of different characters. Use the length of each run to represent the string.

For example, the runs of the binary string "`001011000111101111101001111111`" are $\{00, 1, 0, 11, 000, 1111, 0, 1111, 0, 1, 00, 1111111\}$, so it will be compressed into $\{2, 1, 1, 2, 3, 4, 1, 4, 1, 1, 2, 7\}$.

Input

There are multiple test cases. The first line of input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and m $(1 \leq n \leq 1000,\ 1 \leq m \leq 5 \cdot 10^5)$: the number of runs and the number of queries. The next line contains n integers: $x_1, x_2, \dots, x_n$ $(1 \leq x_i \leq 10^6)$ indicating the length of each run.

Each of the following m lines contains two integers a_i and b_i $(0 \leq a_i, b_i \leq |s|,\ 1 \leq a_i + b_i \leq |s|)$ which means that Professor Zhang wants to know whether there is a substring of s such that the number of occurrences of '`0`' and '`1`' in this substring are exactly a_i and b_i, respectively.

There are no more than 200 test cases, and the total size of the input is at most 20 mebibytes. Additionally, the sum of m in all test cases is at most $2 \cdot 10^6$.

Output

For each test case, print a binary string of length m. The i-th digit must be '`1`' if the answer for the i-th query is "yes", or '`0`' otherwise.

Example

standard input	standard output
3 2 3 3 4 3 0 3 4 1 2 3 4 1 2 3 5 1 4 2 1 3 3 2 12 10 2 1 1 2 3 4 1 4 1 1 2 7 2 1 2 2 2 3 2 4 2 5 4 1 4 2 4 3 4 4 4 5	111 0101 1111101111

Problem I. It's All In The Mind

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Professor Zhang has a number sequence $a_1, a_2, \ldots, a_n$. However, the sequence is not complete and some elements are missing. Fortunately, Professor Zhang remembers some attributes of the sequence:

- For every $i \in \{1, 2, \ldots, n\}$, $0 \le a_i \le 100$.
- The sequence is non-increasing: $a_1 \ge a_2 \ge \ldots \ge a_n$.
- The sum of all elements in the sequence is not zero.

Professor Zhang wants to know the maximum value of $\frac{a_1+a_2}{\sum_{i=1}^{n} a_i}$ among all the possible sequences.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($2 \le n \le 100$, $0 \le m \le n$): the length of the sequence and the number of known elements.

Each of the next m lines contains two integers x_i and y_i ($1 \le x_i \le n$, $0 \le y_i \le 100$, $x_i < x_{i+1}$, $y_i \ge y_{i+1}$) indicating that $a_{x_i} = y_i$.

There are at most 2000 test cases, and the total size of the input is no more than 350 kibibytes.

Output

For each test case, output the answer as an irreducible fraction p/q where p and q are integers, and $q > 0$.

Example

standard input	standard output
2 2 0 3 1 3 1	1/1 200/201

Problem J. Join The Future

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7.5 seconds
Memory limit:	64 mebibytes

Professor Zhang has an array of n integers. He writes down some observations about the array on the paper. Each observation is described by three integers l_i, r_i and s_i, which means that the sum of elements modulo 2 on interval $[l_i, r_i]$ of the array is equal to s_i.

After that, he tries to recover the array only using the above observations. Apparently, there are many such arrays. So, Professor Zhang decides to limit the lower bound and upper bound of each integer in the array.

Given the observations, the lower bounds and the upper bounds, find the number of possible arrays and the lexicographically smallest array.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n \le 40$, $0 \le m \le \frac{n \cdot (n+1)}{2}$): the length of the array and the number of observations.

Each of the next n lines contains two integers x_i and y_i ($0 \le x_i \le y_i \le 10^9$): the lower bound and upper bound of the i-th integer.

Each of the next m lines contains three integers l_i, r_i and s_i ($1 \le l_i \le r_i \le n$, $0 \le s_i \le 1$) denoting the i-th observation.

There are at most 110 test cases, and the total size of the input is at most 30 kibibytes.

Output

For each test case, output the number of possible arrays on the first line. As the value could be very large, print it modulo $10^9 + 7$. Then, output the lexicographically smallest array on the second line. If the number of possible arrays equals to zero, just output "-1" (without the quotes) in the second line.

Example

standard input	standard output
3 3 3 1 10 0 21 3 15 2 2 1 3 3 0 2 3 1 3 0 0 1 1 3 3 4 3 3 1 10 0 21 3 3 2 2 1 3 3 0 2 3 1	660 1 1 4 12 0 1 3 0 -1

Problem K. Keep On Movin

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Professor Zhang has n kinds of characters, and the quantity of the i-th character is a_i. Professor Zhang wants to use all the characters to build several palindromic strings. He also wants to maximize the length of the shortest palindromic string.

For example, let there be 4 kinds of characters denoted as 'a', 'b', 'c', 'd', and let their quantities be $\{2, 3, 2, 2\}$, respectively. Professor Zhang can build ("acdbbbdca"), or ("abbba" and "cddc"), or ("aca", "bbb" and "dcd"), or ("acdbdca and "bb"). The first is the optimal solution where the length of the shortest palindromic string is 9.

Note that a string is called palindromic if it can be read the same way in either direction.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$): the number of kinds of characters. The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^4$).

There are at most 110 test cases, and the total size of the input is at most 6 mebibytes.

Output

For each test case, output an integer denoting the answer.

Example

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

Problem L. La Vie En Rose

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	64 mebibytes

Professor Zhang would like to solve the multiple pattern matching problem, but he only has only one pattern string $p = p_1p_2 \dots p_m$. So, he wants to generate as many pattern strings as possible from p using the following method:

1. select some indices $i_1, i_2, \dots, i_k$ such that $1 \le i_1 < i_2 < \dots < i_k < |p|$ and $|i_j - i_{j+1}| > 1$ for all $1 \le j < k$.
2. swap p_{i_j} and p_{i_j+1} for all $1 \le j \le k$.

Now, for a given a string $s = s_1s_2 \dots s_n$, Professor Zhang wants to find all occurrences of all the generated patterns in s.

Input

The first line contains two integers n and m ($1 \le n \le 10^5$, $1 \le m \le \min(50\,000, n)$): the lengths of s and p, respectively.

The second line contains the string s, and the third line contains the string p. Both strings consist only of lowercase English letters.

Output

Output a binary string of length n. The i-th character must be '1' if and only if the substring $s_is_{i+1} \dots s_{i+m-1}$ is one of the generated patterns. Otherwise, the character must be '0'.

Examples

standard input	standard output
4 1 abac a	1010
4 2 aaaa aa	1110
9 3 abcbacacb abc	100100100

Problem M. Memento Mori

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	64 mebibytes

Professor Zhang has an $n \times m$ matrix consisting of all zeroes. Professor Zhang changes k elements of the matrix into 1s.

Given a permutation p of $\{1, 2, 3, 4\}$, Professor Zhang wants to find the number of such submatrices that:

- The number of 1s in the submatrix is exactly 4.
- Let the positions of the 1s in the submatrix be (r_1, c_1), (r_2, c_2), (r_3, c_3), and (r_4, c_4). Then $r_1 < r_2 < r_3 < r_4$ and $(p_i - p_j) \cdot (c_i - c_j) > 0$ for all $1 \le i < j \le 4$.
- no other submatrices inside the chosen submatrix meet the above two requirements.

Input

There are multiple test cases. The first line of input contains an integer T indicating the number of test cases. For each test case:

The first line contains three integers n, m and k ($1 \le n, m, k \le 2000$): the size of the matrix and the number of 1s. The second line contains four integers p_1, p_2, p_3, p_4 denoting the permutation of $\{1, 2, 3, 4\}$.

Each of the next k lines contains two integers r_i and c_i ($1 \le r_i \le n$, $1 \le c_i \le m$): the position of the i-th 1. No two 1s will be in the same position.

There are at most 250 test cases, and the total size of the input is at most 250 kibibytes.

Output

For each test case, output a single integer: the number of submatrices which meet all the requirements.

Example

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