Problem A. Apples

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

There are n apple trees growing in the garden.

If at night some tree has no apples, then one apple appears on this tree. During each of m consecutive days Alice goes to the garden. Each day she chooses some rectangle with sides parallel to coordinate axis and collects apples from all trees in this rectangle.

Your task is to calculate how many apples Alice will have after m days.

Input

In first line of input, there is an integer T ($1 \le T \le 5$) indicating the number of test cases.

The first line of the test case contains two integers n ($0 < n \le 10^4$) and m ($0 \le m \le 5 \cdot 10^4$), separated by a blank, where n is the number of apple trees and m is number of days Alice was in the garden.

Each of the following n lines contains two integers x and y ($0 \le x, y \le 10^4$) indicating that there is an apple tree located at (x, y). Then m lines follow. Each of them contains four integers l, r, b, t ($0 \le l \le r \le 10^5$, $0 \le b \le t \le 10^5$) separated by spaces. $[l_i, r_i] \times [b_i, t_i]$ corresponds to the rectangle Alice chosen at the i-th day.

Output

Print the total number of apples Alice will collect for all m days.

Example

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

Problem B. Black and White Map

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given a closed polyline on the infinite plane; it is known that any point of the plane lies on no more than two edges of the polyline. A polyline divides the plane to the *regions*; border of regions is formed by the polyline's edges and its parts; an exterior of the polyline is considered an infinite region. We say that two regions are *neighbors*, if there exists a segment of positive length that belongs to the border of both regions.

Now we want to choose some regions and paint them black, in such a way that for any two neighbors exactly one of them is black. Calculate the minimum possible total area of the black regions.

Input

First line of the input contains one integer n — number of the edges of the polyline ($3 \le n \le 500$).

Each of the next n lines contain two integers x_i and y_i — coordinates of one vertice of polyline. Vertices with coordinates (x_i, y_i) and (x_{i+1}, y_{i+1}) are connected by the edge, same as vertices (x_1, y_1) and (x_n, y_n), $-1000 \le x_i, y_i \le 1000$).

Output

Print miniumum total area of black regions with absolute error 10^{-4} or better. If it is impossible to build such a coloring, print -1 instead,

Examples

standard input	standard output
3 0 0 1 1 1 0	0.5000
4 0 0 2 0 2 6 4 4	6.0000

Problem C. Chips and Matches

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Mary found a pile of chips and matches. The chips are arranged in a rectangular grid $R \times C$.

For two chips we say that they are adjacent if they are in the same or adjacent row, and in the same or adjacent column. Each chip can therefore have up to 8 neighbor chips.

Then Mary sets up matches between some adjacent chips. We say two chips are *connected* if there is a path that runs from the first to the second chip through one or more matches. Every pair of chips is connected and no two matches cross each other.

In the example below we see nine chips and matches that link them. In order to uniquely record match positions, for each chip, we will determine the 4-bit number as follows:

- The bit 0 ($2^0 = 1$) indicates the match to the top right neighbor chip.
- The bit 1 ($2^1 = 2$) indicates the match to the right neighbor chip.
- The bit 2 ($2^2 = 4$) indicates the match to the bottom right neighbor chips.
- The bit 3 ($2^3 = 8$) indicates the match to the bottom neighbor chip.

The positions of the matches can finally be written as the $R \times C$ matrix of the hexadecimal digits we get by converting the corresponding 4-bit numbers.

Write a program that will calculate number of ways Mary can move one match to another position so that each pair of chips is still connected and that the matches still do not intersect with each other.

Input

First line of the input contains two integers R and S ($2 \leq R, S \leq 100$) — number of rows and columns in the grid.

Each of next R lines contains S hexadecimal digits (digits '`0`' - '`9`' denote numbers from 1 to 9, digits '`A`' - '`F`' denote numbers from 10 to 15).

Output

Print one integer — number of ways to move one match such as each pair of chips is still connected and matches still do not intersect each other.

Example

standard input	standard output
2 2 20 30	5
3 2 88 88 20	20
3 3 8E8 FA8 220	43

Problem D. Drink from Mimisbrunnr!

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Mimisbrunnr is a magical well associated with Mimir who is a wisdom deity in Norse mythology. It is said that Odin drank from Mimisbrunnr and obtained wisdom and intelligence. As an ICPC contestant, you also need wisdom and intelligence to win the trophy, so you ask Mimir for drinking from Mimisbrunnr.

Mimir is wise, but he is very old. Mimir constructed the well at a perfect location, but he forgot where it is. "Odin sacrificed his right eye to exchange the right to drink. However, you are just a poor programmer, you may submit a source code to exchange my permission. If your program computes the location of Mimisbrunnr, then you may drink from it and obtain what you need to win the ICPC." Mimir tells you.

The spring from Mimisbrunnr is a mixture of water from n sources located at $(x_1, y_1), \ldots, (x_n, y_n)$. Mimir built the magical well Mimisbrunnr at a location such that the total distances from there to n sources is minimized. That is, $\sum_{i=1}^{n} \sqrt{(x - x_i)^2 + (y - y_i)^2} \leq \sum_{i=1}^{n} \sqrt{(x' - x_i)^2 + (y' - y_i)^2}$ for any real numbers x' and y'. If there are multiple instances achieving the minimum, Mimir would choose the one minimizing x. If there still are multiple instances with the same x, then Mmir would pick the one minimizing y.

It is time to obtain wisdom and intelligence equivalent to Odin's. Submit your code, then drink from Mimisbrunnr!

Input

For each test case, the first line contains an integer n ($n \leq 10^4$) indicating the number of sources of Mimisbrunnr. The i-th of the following n lines contains two integers x_i and y_i which indicate the i-th source is located at (x_i, y_i). In all test cases, you may assume that $x_i, y_i \in [-10^4, 10^4]$.

Output

For each test case, output x and y in one line. Use a blank to separate them. Any error no more than 5×10^{-3} is acceptable.

Example

standard input	standard output
2 1 0 1 1	1.00 0.00
3 0 0 1 2 2 0	1.00 0.58

Problem E. Earthquake and Permutation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Little Dasha has a permutation P of integers from 1 to n written on a board. Each day, she swaps some elements in P to form a new permutation P', and then tries to find the longest increasing subsequence (LIS) of P'. She believes that she can eventually find a faster algorithm for calculating LIS.

One day after an earthquake, some of the numbers in the permutation P dropped off the board. Little Dasha soon wondered, if she puts all the numbers back in a certain way, what will be the largest possible length of the LIS of the recovered permutation?

Input

The first line of the input contains an integer n ($1 \le n \le 10^5$) indicating the length of the permutation. The next line contains a incomplete permutation $a_1,\ a_2,\ \ldots,\ a_n$. If $a_i = 0$, that means the number originally at the i-th position has dropped.

Output

Print the maximum possible length of the LIS of the recovered permutation.

Examples

standard input	standard output
6 0 0 0 0 0 0	6
12 1 0 3 0 5 0 7 0 9 0 11 0	12
9 3 0 0 0 7 8 9 0 0	7

Problem F. Find the Cheapest Covering

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Lets define a *route* in directed graph from vertice v_0 to vertice v_k as the sequence of vertices and edges

$$v_0,\ e_1,\ v_1, \ldots, v_{k-1},\ e_k,\ v_k$$

where edge e_i connects vertices v_i and v_{i-1}. *Cost of the vertex* is defined as the number of outgoing edges from this vertex, and *cost of the route* is defined as the sum of costs of all vertices in this route.

Consider directed graph with n vertices and m edges with next property: for any edge e there exists the route from vertex 1 to vertex n that contains this edge. We define the *covering* as a set of routes from vertex 1 to vertex n, such that each edge is a part of at least one of these routes. Total cost of the covering is defined as the sum of costs of all routes in the covering.

Your task is to find the covering of minimum total cost.

Input

First line of the input file contains two integers n and m ($2 \le n \le 200,\ 1 \le m \le 1000$) — number of vertices and edges of the given graph, respectively. Each of next m lines contains description of one edge — starting vertice a_i and ending vertice b_i. Note that there may be loops and multiple edges ($1 \le a_i, b_i \le n$).

Output

Print the covering with minimum total cost in the next form: in first line print the cost, in second line print number of routes l, in next l lines print the routes, one per line. For each route first print integer l — number of vertices in the route, then l integers — vertices in the route, listed in order defined by this route.

Examples

standard input	standard output
4 4 1 2 2 3 3 2 3 4	7 1 6 1 2 3 2 3 4
4 7 1 2 1 4 2 3 2 3 3 2 3 3 3 4	17 2 7 1 2 3 2 3 3 4 2 1 4

Problem G. Grid

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given the grid $N \times M$. Each cell of the grid is painted in one of two colors: black or white.

In one turn is allowed to choose a cell and reverse colors on the same horizontal and the same vertical with selected cell (including this cell), i.e. black cells are replaced with white ones and vice versa.

Given initial coloring check if it is possible to make all cells on the grid white.

Input

First line of the input contains two space-separated integers N and M ($1 \leq N, M \leq 50$).

Then N lines follow — description of the initial position. Each line contains a string of length M, consisting of letters '`B`' if initially corresponging cell is black and '`W`' otherwise.

Output

If it is impossible to paint all cells white, print "`No`". Otherwise in first line print "`Yes`", in second — number of turns, each of next lines must contain two integers — row and column of cell, selected at this turn.

Examples

standard input	standard output
2 3 WWB BBB	Yes 1 2 3
3 3 WWW WBW WWW	No

Problem H. Hackers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Young hackers Alice, Bob and Cathy had to take the exam yesterday. The exam consisted of n multiple choice questions. There is one point for each question, and each question had four choices '`A`', '`B`', '`C`', and '`D`'. Students submitted their answers sheets for computer scoring.

After the exam they are found that they submitted a wrong answer for every question. They decide to crack into the teacher's computer to modify the correct answers. They aims to maximize the minimum of their scores.

For example, assume there were 3 questions in the exam, Alice answered `ABC`, Bob answered `BCD`, and Cathy answered `CDA`, and the correct answers were `DAB`. They answered wrongly to all questions, but when they modify the answers to `CCC`, each of them will "correctly" answer at least one question in the exam.

Given scores of each student, find out how to modify the standard answers to maximize the minimum of their scores.

Input

The first line contains an integer T ($1 \le T \le 100$) indicating the number of test cases.

Each test case consists of four lines. The first line contains an integer n ($1 \le n \le 3 \cdot 10^4$) indicating the exam consisted of n questions. The following three lines represent Alice's, Bob's and Cathy's answers, respectively. Each of these lines contains a string of n characters between '`A`' and '`D`' inclusively, and the i-th character is the answer to i-th question written on the sheet.

Output

For each test case print one line containing an integer indicating the maximum possible value of the minimum of their scores.

Example

standard input	standard output
2 3 ABC BCD CDA 4 AABC AACD BBAA	1 2

Problem I. Indiana Jones and the Maze

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Indiana Jones is walking into the maze and collecting the values. In this case maze is a tree, i.e. a connected graph that contains N vertexes and $N-1$ edges. Indiana wants to get maximum values by walking from a vertex A and stopping at a vertex B.

Indiana Jones is allowed to choose a vertex A (the value of A can not be 0) and a vertex B by himself. Walking from A and stopping at B, he must collect all the values on the road. Each vertex has a value. Indiana Jones tries to get values as large as he can. But he can never go back to any vertex he has passed.

However, there is a special way to calculate total values. Let's assume that Indiana Jones has passed M vertexes from A to B. During the travel, Indiana Jones has successively collected M values worths W_i $(0 \le i < M)$. Vertex A has a value worth W_{M-1}. The next vertex on the road has a value worth W_{M-2}, ... At last, vertex B has a value worth W_0.

Given an integer P, the total value Indiana Jones will collect is calculated by the formula

$$MAX = \sum_{i=0}^{M-1} (W_i \times P^i).$$

It is guaranteed that W_i $(0 \le i < M)$ are less than P. The vertex A and B you choose can be same. But the value of A can not be 0. Output MAX modulo $(10^9 + 7)$.

Note that you need to make sure MAX as large as possible but NOT make sure the remainder as large as possible. And then, output value of each vertex (stating from vertex A) on the road in the best case.

Input

The first line contains an integer $T(1 \le T \le 200)$, indicating the number of test cases.

For each test case, the first and second line contain two integers N $(1 \le N \le 10^4)$ and P $(2 \le P \le 10^9)$, indicating the number of vertexes and the integer P.

Each of the following $N-1$ lines contains two integers a and b $(1 \le a, b \le N, a \ne b)$, indicating that there is an edge connecting vertex a and vertex b.

The following line contains N integers W_i $(0 \le W_i < P, \sum W_i > 0)$, the value of each vertex. It is guaranteed that at least one of W_i not equal to zero.

You can assume that sum of all N in the test cases does not exceed 1.3×10^6.

Output

For each test case, print two lines. First must contain the maximum value of treasures Indiana Jones can collect modulo $(10^9 + 7)$.

In the second line print the values of each vertex from vertex A to vertex B.

Example

standard input	standard output
2 8 2 1 2 2 3 3 4 4 5 2 6 6 7 7 8 1 0 0 0 0 0 0 0 9 923456789 1 2 2 3 1 4 4 5 1 6 6 7 1 8 8 9 1 2 3 2 0 2 0 2 3	16 1 0 0 0 0 90485756 3 2 1 2 3

Problem J. Jaguars

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Jennifer studies biology. Now she is writing a paper about behaviour of jaguars. So Jennifer decided to do some experiments with them.

There is a flat desert and N locations, numbered from 1 to N, on it. The i-th location's coordinates are (x_i, y_i), and no two locations coincide. After a period of observation, Jennifer has noticed that:

1. When a jaguar is at location p, it always moves to another location which is closest to p. If there are multiple locations with minimum distance from p, it moves to the one with smaller x-coordinate value. If there is still a tie, the one with smaller y-coordinate value is chosen. When a jaguar moves from a location to another one, it always moves along the segment connecting them.
2. Jaguar never stops, that is, when a jaguar arrives to a location, it moves to the next location immediately. So, of course, ajaguar may visit a location infinitely many times.
3. All the jaguars move at the same speed.

Izabel now is curious if, given two different locations and two jaguars start to move from these two locations simultaneously, will they meet at some time during their moving?

Note that all the jaguars and locations can be considered as points.

Input

The first line of the input contains two integers N and Q ($2 \le N \le 10^5$, $1 \le Q \le 10^5$), indicating the number of locations and the number of queries respectively.

The following N lines, each contain two integers x_i and y_i ($-10^9 \le xi, yi \le 10^9$), indicating coordinates of the i-th location. The following Q lines each contain two integers i and j ($1 \le i, j \le N$, $i \ne j$), indicating the number of the two given locations.

Output

For each query, output "`hi`" in a line if the two jaguars will meet; otherwise output "`oops`".

Example

standard input	standard output
2 1 1 1 0 2 1 2	hi
5 2 1 1 3 3 4 4 0 -3 0 -4 1 3 2 4	hi oops

Problem K. Keywords

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

You need to consider all keywords of length L consisting of characters from the alphabet A and calculate average number of distinct non-empty substrings in those keys.

Input

First line of the input contains two integers L and N N ($1 \le L \le 11$, $1 \le N \le 52$) — length of the keywords and number of characters in A. Second line contains N space-separated English letters — the alphabet A. Note that lowercase and uppercase letters are considered distinct.

Output

Print one integer — average number of distinct non-empty substrings in the keyword. Answer must be printed in form p/q, where $p \ge 0$, $q > 0$, p and q are integers and $gcd(p, q) = 1$.

Examples

standard input	standard output
1 5 a A b B c	1/1
3 1 z	3/1
2 2 z x	5/2

Note

In the first sample all 5 possible keywords have the length 1 so they have only one substring each.

In the second sample exists only one password "zzz" and three distinct substrings "z", "zz" and "zzz".

In the third sample possible substrings are "xx", "zx", "xz" and "zz".

Problem L. Lazy Kids

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Every weekend little Laura and Leonard play in the park. Because they are too lazy, for each time, one of them has to go for food for both. They agree to apply the following procedure to determine who is on the duty today.

- Put n red balls and m white balls in a paper bag.
- Kids draw a ball from the bag in turns. Note that once a ball is drawn, it is removed from the bag.
- The person who draws a red ball first has to buy drinks.

Leonard lets Laura perform the first draw every week. One day, Laura suddenly wonders what is the probability that she draws a red ball first. Can you help her to calculate it? You may assume when every ball has the same probability to be drawn.

Input

The first line contains a positive integer T , $T \leq 1770$, indicating the number of test cases. Each test case has one line containing two positive integers n and m where $1 \leq n \leq 59$, $1 \leq m \leq 59$ and $n + m \leq 60$. n is the number of red balls, and m is the number of white balls.

Output

For each test case, output one line containing the answer represented by an irreducible fraction.

Example

standard input	standard output
2 1 1 1 2	1/2 2/3