Counting D-sets

- SUBMIT
- ALL SUBMISSIONS

Problem code: CNTDSETS

Read problems statements in Mandarin Chinese and Russian.

Chef likes points located at integer coordinates in a space having **N** dimensions. In particular, he likes sets of such points having diameter **exactly equal to D** (called D-sets). The diameter of a set of points is the maximum distance between any pair of points in the set. The distance between two points **$(a_1,a_2,...,a_N)$** and **$(b_1,b_2,...,b_N)$** is **$\max\{|a_1-b_1|, |a_2-b_2|, ..., |a_N-b_N|\}$**.

Chef would like to know how many D-sets exist. However, he soon realized that, without any extra constraints, there is an infinite number of D-sets. Thus, he would only like to count the number of classes of D-sets, such that any two D-sets which belong to the same class are equivalent under translation. To be more precise, two D-sets **X** and **Y** are considered equivalent (and belong to the same class) if:

- they contain the same number of points **AND**
- there exists a tuple of **N** integer numbers **$(t_1, ..., t_N)$** such that by translating each point of **X** by the amount **t_i** in dimension **i** (**$1 \le i \le N$**) we obtain the set of points **Y**.

Let's consider **N=2**, **D=4** and the following sets of points **X={(1,2), (5,5), (4,3)}** and **Y={(2,5), (5,6), (6,8)}**. Let's consider now the tuple **(1,3)**. By translating each point of **X** by the amounts specified by this tuple we obtain the set **{(2,5), (6,8), (5,6)}**, which is exactly the set **Y**. Thus, the two sets **X** and **Y** are equivalent and belong to the same class.

Help Chef find the number of classes of D-sets **modulo 1000000007** (**10^9+7**).

Input

The first line of input contains the number of test cases **T**. Each of the next **T** lines contains two space-separated integers describing a test case: **N** and **D**.

Output

For each test case (in the order given in the input), output the number of classes of D-sets **modulo 1000000007**.

Constraints

- **$1 \le T \le 10$**
- **$1 \le N \le 1000$**
- **$1 \le D \le 1000000000$** (**10^9**)

Example

Input:

5
1 10
2 1

2 10
3 1
3 3

Output:
512
9
498134775
217
548890725

Explanation

Example case 1:

When **N=1** all the points are arranged in a line. In order to have a diameter equal to **10** each D-set must contain two points at distance **10**. Between two such points there may be up to **9** points which may belong to the D-set or not. Thus, there are **2^9=512** classes of D-sets.

Example case 2:

There are **9** classes of D-sets. One D-set from each class is given below:

- {(0,0), (0,1)}
- {(0,0), (1,0)}
- {(0,0), (1,1)}
- {(0,1), (1,0)}
- {(0,0), (0,1), (1,0)}
- {(0,0), (0,1), (1,1)}
- {(0,0), (1,0), (1,1)}
- {(0,1), (1,0), (1,1)}
- {(0,0), (0,1), (1,0), (1,1)}

Counting The Important Pairs

- [SUBMIT]
- [ALL SUBMISSIONS]

Problem code: TAPAIR

Read problems statements in Mandarin Chinese and Russian.

*[**a simple graph** is an undirected graph that has no loops (edges connected at both ends to the same vertex) and no more than one edge between any two different vertices.]*

-Retrieved from http://en.wikipedia.org/wiki/Graph_(mathematics)

The road system at Byteland can be seen as a simple graph in which the vertices represented by the cities and the edges represented by the roads connected two different cities. Besides, the road system is connected it means people can travel between any two cities. It is decided by the president that two roads will be concurrently upgraded in the next month. Since it may take a couple of weeks to get the work done, two roads should be chosen so that the connectivity of the system is hold during the up-gradation process. More clearly, people still can go between any pair of cities without using that two roads.

Which roads should be upgraded is not decided yet. There may be no way to choose such a two roads or may be many ways. Your mission is counting how many pair of roads **cannot** be chosen. Notice that two roads must be different and we consider only un-ordered pair ((1, 2) is the same as (2,1)).

Input

The first line of the input contains two integers **N M** which are the number of cities and roads respectively. Each line in the next **M** contains two integers **u v** which means there is a road connects the**u**th city to the **v**th city.

Output

Output a single line contains the number of way that two roads **cannot** be chosen for upgradation.

Constraints

- **1 ≤ N ≤ 100,000**
- **1 ≤ M ≤ 300,000**
- **1 ≤ u, v ≤ N**
- **u ≠ v**
- There is at most one road between any two cities.

Example

Input:
5 6
1 2
2 3

1 3
3 4
4 5
3 5

Output:
6

Explanation

There are 5 cities and 6 roads (numbered from 1 to 6). There are 6 pairs of roads cannot be chosen which are

(1, 2), (1, 3), (2, 3), (4, 5), (4, 6) and (5, 6).

Byteland Tours

- SUBMIT
- ALL SUBMISSIONS

Problem code: TOURBUS

Read problems statements in Mandarin Chinese and Russian.

As part of a new program to attract tourists to Byteland, Chef has been placed in charge of organizing a series of bus tours around the city. There are **N** scenic locations (points on the plane) in Byteland (numbered **0** through **N-1**), and **M** bidirectional scenic roads, each of which connects two scenic locations via a straight line. Chef wishes to organize the tours according to the following rules:

- A tour is defined as sequence of locations, where a tour bus begins at the first location, and moves directly from each location to the next location in the sequence until it reaches the last location. There must be a scenic road between consecutive locations.
- A tour does not have to start and end at the same location, but it is allowed.
- A tour cannot visit the same location more than once, with the possible exception of the start/end location.
- A tour cannot cross itself. That is, none of the roads belonging to a tour may intersect.
- Each road must be visited by exactly one tour.

Setting up tours is expensive, so Chef wishes to minimize the total number of tours. Optimal solutions are not required, but solutions that use fewer tours will score more points (see the scoring section below). However, your solution must use at most **(N+M)/2** tours or else it will be judged Wrong Answer.

Input

Input will begin with a line containing an integer **N**. Following this will be **N** lines containing the locations of the **N** scenic locations. Each location is given as a pair of integers, with the **i**-th line giving **x_i** and **y_i**, the respective **x** and **y** coordinates of the **i**-th scenic location. Following this will be **N** lines containing **N**characters each. The **i**-th character of the **j**-th line will be 'Y' if there is a scenic road connecting locations**i** and **j**, and 'N' otherwise.

Output

First, print an integer **K**, the number of tours. Then print **K** tour descriptions. Each tour description should consist of an integer **L**, the length of the tour (in roads), followed by **L+1** integers, the ordinal numbers of the locations on the tour, in order.

Scoring

Your score is **K*N/M**. Lower scores will earn more points.

We have 50 official test files. You must correctly solve all test files to receive OK. During the contest, your overall score is the sum of the scores on the first 10 test files. After the contest, all solutions will be rescored by the sum of the scores on the rest 40 test files. Note, that public part of the tests can not contain some border cases.

Constraints

- 20 ≤ **N** ≤ 50
- 0 ≤ **x_i**, **y_i** ≤ 100
- No three points will be collinear.
- The i-th character of the i-th row of the road descriptions will be 'N'.
- The i-th character of the j-th row of the road descriptions will equal the j-th character of the i-th row.
- Each city will be connected to at least one road.

Sample Input

```
6
2 3
10 0
10 7
3 7
9 8
2 1
NNNYNY
NNYYNN
NYNYYN
YYYNYN
NNYYNY
YNNNYN
```

Sample Output

```
3
4 3 0 5 4 2
1 4 3
3 1 2 3 1
```

This sample output would score **3*6/8 = 2.25** points.

Test Case Generation

N will be chosen randomly and uniformly between 20 and 50, inclusive. Each point's coordinates are chosen randomly and uniformly between 0 and 100, inclusive. If two points coincide or three points are collinear, the process is restarted (with the same value of **N**).

A real value **P** is chosen randomly and uniformly between 0.3 and 1.0. Between each pair of cities a road exists with probability **P**. If some city has no roads, the process is restarted (with the same value of **P**).