Problem A
Arranging Marbles

The game of marbles (or shooting marbles) is very popular for kids in many countries. In Vietnam, the most classic way to play this game is to shoot your marbles at opponents' ones to get it.

Hieu is a truly professional player and has won many games, which brought him a collection of rare colorful marbles. His collection consists of n^2 marbles which are colored in at most $n+1$ different colors. The marbles are numbered from 1 to n^2, and the colors are numbered from 1 to $n + 1$ (inclusive). Hieu wants to put his n^2 marbles in n boxes so that the following conditions hold:

- Each box consists of exactly n marbles;
- The marbles in each box are colored in at most 2 colors.

Given the color of all Hieu's marbles, your task is to help him arrange them.

Input

- The first line contains an integer n ($1 \leq n \leq 1\,000$).
- The second line contains n^2 integers $c_1, c_2, \cdots, c_{n^2}$ ($1 \leq c_i \leq n + 1$), where c_i denotes the color of the i-th marble. There might be a color which never appears in any marbles.

Output

If it is impossible to arrange Hieu's marbles, print a single word `NO`. Otherwise:

- The first line contains the word `YES`.
- The second line contains n^2 integers $b_1, b_2, \ldots, b_{n^2}$ ($1 \leq b_i \leq n$) where b_i denotes the box which has the i-th marble.

If there are multiple solutions, you can output any of them.

Sample Input 1

```
3
1 1 1 1 3 3 3 3 3
```

Sample Output 1

```
YES
1 1 2 2 1 2 3 3 3
```

Problem B
Beautiful Board

Minh has a board of size $R \cdot C$ (R rows and C columns). Each cell of the board contains an upper-case letter.

Minh believes that the board is beautiful if and only if it has 2 axes of symmetry: a vertical one and a horizontal one. In other words, for each of the R rows, the string in that row must be a palindrome; and for each of the C columns, the string in that column must be a palindrome.

To make the board beautiful, Minh can apply a series of operations. Each operation can be any of the following:

- Select a cell (i, j) and change the character to the next character in the alphabet. The next character of Z would be A.
- Select a cell (i, j) and change the character to the previous character in the alphabet. The previous character of A would be Z.

Your task is to find the minimum number of operators Minh needs to achieve his goal.

Input

The input starts with a line containing 2 integers R and C denoting the size of the board ($1 \leq R, C \leq 50$). Then R lines follow, each contains a string with exactly C upper-case characters.

Output

Print the minimum number of operations needed to transform the original board into a beautiful board.

Sample Input 1	Sample Output 1
3 3 AAB AAA BAB	1

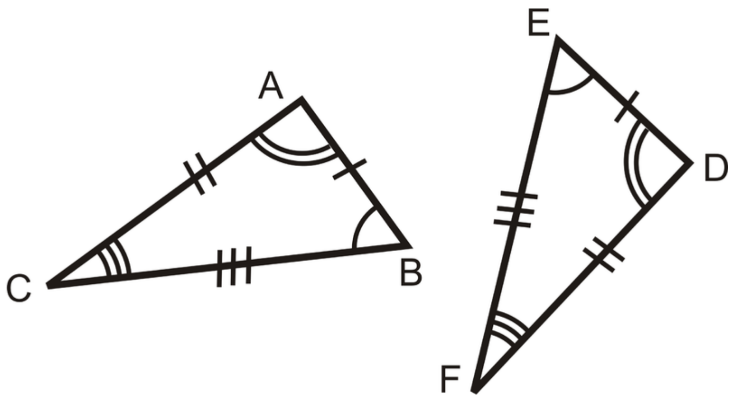
Sample Input 2	Sample Output 2
2 4 LOVE ICPC	33

Problem C
Congruent Triangles

In geometry, two triangles are considered *congruent* iff they have exactly the same size and shape. In other words, all pairs of corresponding interior angles are equal in measure, and all pairs of corresponding sides have the same length. More formally, triangle ABC is congruent to triangle DEF, mathematically written as $\triangle ABC \cong \triangle DEF$, if and only if all six below statements are true:

- $AB = DE$
- $BC = EF$
- $CA = FD$
- $\angle ABC = \angle DEF$
- $\angle BCA = \angle EFD$
- $\angle CAB = \angle FDE$

Note that while stating congruence of triangles, the order of vertices matters. For example, by stating $\triangle ABC \cong \triangle DEF$, we means that the side AB corresponds to the side DE, the angle $\angle BCA$ corresponds to the angle $\angle EFD$,... Therefore, in the below figure, two statements $\triangle ABC \cong \triangle DEF$ and $\triangle ACB \cong \triangle DFE$ are true; while $\triangle ABC \cong \triangle FED$ is false since $AB \neq FE$.



In this problem, you are given n points on the Cartesian plane, denoted as $P_1, P_2, \ldots, P_n$. The coordiates of point P_k is (x_k, y_k). You are about to count the number of pairs of congruent triangles, whose vertices are 6 distinct points among the given ones.

Formally, you should count the number of tuples of indices $(i_1, i_2, i_3, j_1, j_2, j_3)$ satisfying all the below conditions:

- $1 \leq i_1, i_2, i_3, j_1, j_2, j_3 \leq n$
- $i_1 < i_2 < i_3$
- 6 indices $i_1, i_2, i_3, j_1, j_2, j_3$ are pairwise distinct.
- Triangle $P_{i_1}P_{i_2}P_{i_3}$ is non-degenerate (in other words, it has positive area).
- $\triangle P_{i_1}P_{i_2}P_{i_3} \cong \triangle P_{j_1}P_{j_2}P_{j_3}$

Input

The first line contains a single integer n ($6 \le n \le 100$) — denoting the number of given points.

Among the last n lines, the k-th one contains two integers x_k and y_k ($0 \le |x_k|, |y_k| \le 10^9$) denoting the coordinates of point P_k.

It is guaranteed that all given points are pairwise distinct.

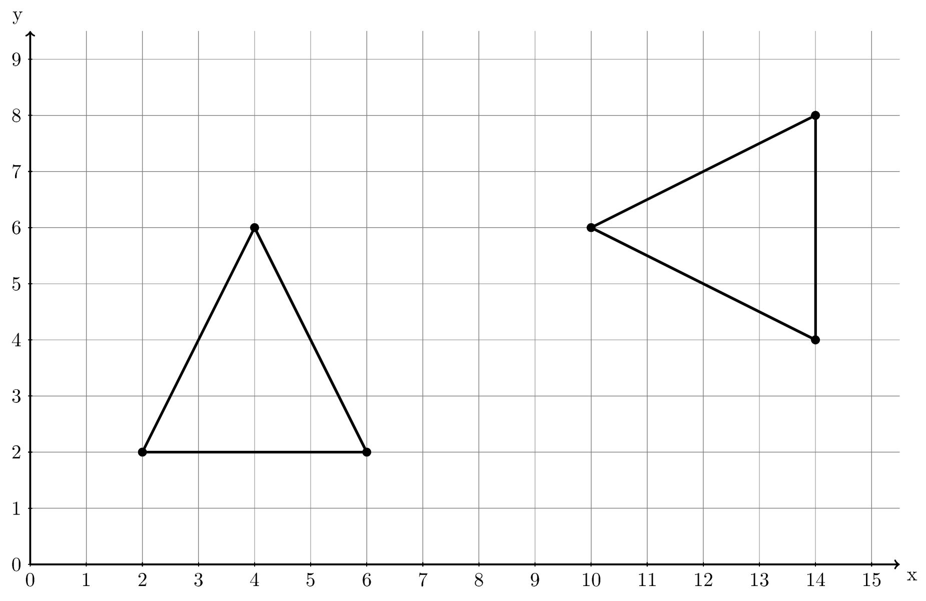
Output

Print a single integer — the number of tuples satisfying all the above conditions.

Explanation of the sample

The below figure demonstrates the above sample. The 4 valid tuples are:

- $(1, 2, 3, 4, 5, 6)$
- $(1, 2, 3, 4, 6, 5)$
- $(4, 5, 6, 1, 2, 3)$
- $(4, 5, 6, 1, 3, 2)$



Sample Input 1	Sample Output 1

6
2 2
4 6
6 2
14 8
10 6
14 4
	4

Problem D
Distinctive Tours

The city of Hanoi has n sightseeing spots, which are numbered from 1 to n, inclusive. There are m two-way roads connecting these spots. These roads form a simple graph: no two roads connect the same pairs of spots, and no road connects a spot to itself.

Each road is decorated with a different type of tree. Hanh is a tree-lover and he wants to create a set of k tours which satisfy:

- Each tour is a cycle of length $t (t > 2)$ that passes through t sightseeing spots $p_1, p_2, \cdots, p_t$. More precisely,
  - for all i that $1 \le i \le t-1$, p_i and p_{i+1} must be directly connected by some road;
  - p_t and p_1 must also be directly connected by some road;
  - all p_i are pair-wise distinct.
- Each tour must have at least one road which does not belong to any of the other $k-1$ tours.

Hanh realizes that it might not be possible to create such a set using the current road network. Therefore, he wants to add some two-way roads so that:

- The new set of roads (including the added and the original ones) still form a simple graph: no two roads connect the same pairs of spots, and no road connects a spot to itself.
- The number of added roads should be minimal.

Your task is to help Hanh add new roads and create a k-tour set.

Input

- The first line contains three integers n, m and k ($3 \le n \le 50, 0 \le m \le \frac{n\cdot(n-1)}{2}, 0 \le k \le 2\,000$).
- In the next m lines, each contains two integers u and v ($1 \le u, v \le n$) meaning that initally there is a road connecting two spots u and v. It is guaranteed that these m roads form a simple graph.

Output

If it is impossible to create a k-tour set no matter how Hanh adds new roads, print a single line containing the word `NO`. Otherwise:

- The first line contains the word `YES`.
- The second line contains a single integer w — the minimal number of added roads.
- In the next w lines, each contains two integers x and y ($1 \le x, y \le n$) meaning that a road connecting two spots x and y should be added.

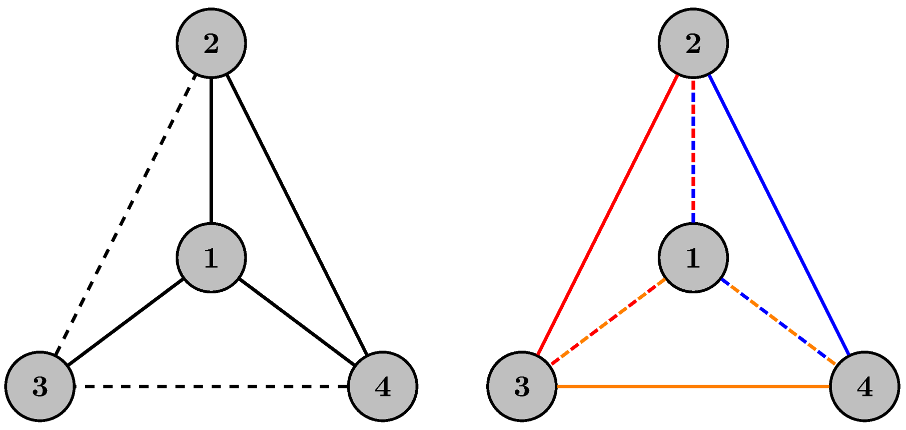
- In the last k lines, each describes a tour in the below format:
  - The first integer is t — the number of spots of the tour.
  - The last t integers are $p_1, p_2, \ldots, p_t$ — the spots of the tour.

If there are multiple optimal solutions, you can output any of them.

Explanation of the samples

The figures below shows the first sample.

- On the left, the original roads are represented by solid segments, the added roads are represented by dashed segments.
- On the right, there are 3 tours: red, blue and orange. Roads are colored with tours that used them. You can see that each tour has one road that does not belong to the other tours: $(2, 3)$ for red, $(2, 4)$ for blue and $(3, 4)$ for orange.



In the second sample, the current roads form a complete graph, so you can not add any roads. You can only create 1 tour using all these current roads.

Sample Input 1

```
4 4 3
1 2
1 3
1 4
2 4
```

Sample Output 1

```
YES
2
2 3
3 4
3 1 2 3
3 1 2 4
3 1 3 4
```

Sample Input 2

```
3 3 3
1 2
1 3
2 3
```

Sample Output 2

```
NO
```

Problem E
Even Paths

Last year, Arthur participated in the Vietnamese Robotic Olympiad with a malfunctioning robot. This year, he decided to participate one more time.

The challenge for contestants this year is much harder. The playing field is a square grid, with the bottom left corner at $(0, 0)$ and the upper right corner at (M, M). In each of the $(M + 1)^2$ grid points, there is a light. Initially, exactly N lights are turned on. The robot is allowed to have multiple runs and its target is to turn off all the lights.

A valid run of the robot is a sequence of grid points $(px_0, py_0), (px_1, py_1), \cdots, (px_k, py_k)$ where:

- these $k + 1$ points should be inside the playing field (formally $0 \le px_i, py_i \le M$ for every $0 \le i \le k$);
- these points are pairwise distinct;
- two consecutive grid points must be neighbor, i.e. $|px_i - px_{i+1}| + |py_i - py_{i+1}| = 1$ for every $0 \le i < k$;
- the length of the path (k) must be an **even** number.

According to the rules of the Olympiad, in a run, whenever the robot passes through a grid point, it can choose to turn off the light there. However, as Arthur's robot is buggy again, his robot decides to **always switch the light** (on to off, off to on) whenever the robot passes through a grid point. In other words, all $k + 1$ lights at $(px_0, py_0), (px_1, py_1), \cdots, (px_k, py_k)$ are switched.

With this buggy robot, it is even hard just to turn off all the lights. Your task is to help Arthur to accomplish that. He will be extremely thankful if you manage to turn off all the lights. The number of runs does not need to be minimized, but it should not exceed $10\,000$.

Input

The input starts with an integer T — the number of test cases ($T \le 100\,000$). The T test cases follow with the format:

- The first line consists of 2 integers M and N ($1 \le M \le 50$, $0 \le N \le (M + 1)^2$).
- In the next N lines, the i-th one consists of 2 integers x_i, y_i presenting the location of a light which is turned on at the beginning. All these locations are pairwise distinct.

It is guaranteed that the sum of N in all the test cases does not exceed $100\,000$.

Output

For each test case in the input:

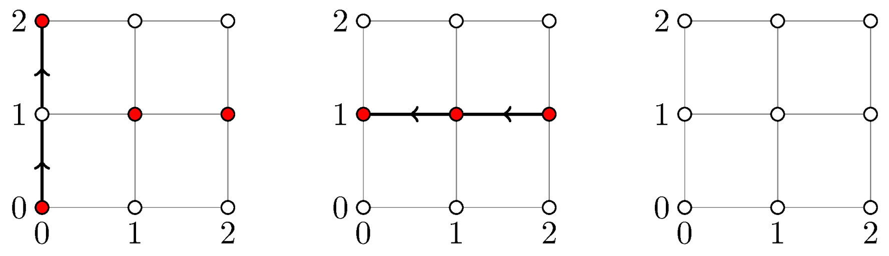
- If it is impossible to turn off all the lights using at most $10\,000$ runs, print -1.

- Otherwise, you should print P $(0 \leq P \leq 10\,000)$ — the number of paths on the first line, followed by P lines describing P paths. Each line should start with an integer k — the length of the path, followed by $(k + 1)$ pairs of integers px_i, py_i $(0 \leq i \leq k)$ representing coordinates of points in the path.

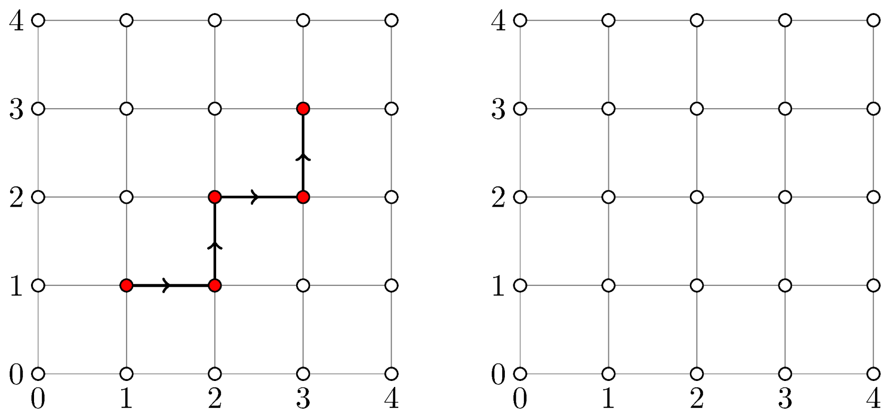
If there are multiple solutions, you can output any of them.

Explanation of the sample

The following 3 images show the first sample: the original grid and the first run, the grid after the first run and the second run, the grid after the second run. The white circle denotes a light turned off, a red circle denotes a light turned on.



The following 2 images show the second sample: the original grid and the first run, the grid after the first run. The white circle denotes a light turned off, a red circle denotes a light turned on.



Sample Input 1

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

Sample Output 1

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

Problem F
Final Ranking

This year, due to the COVID-19 situation in Vietnam, The ICPC Vietnam National Contest will be held after 3 months of delay. Being the host of the contest, all the students of FPT university want to qualify for their teams and to represent their university in the contest. They worked very hard to sharpen their problem-solving skills. FPT University's coach organized a series of individual contests to prepare his n students and to be able to select the best team. After all contests, the rank of a contestant equals *the number of contestants that have higher score* plus 1. This is an example of a final ranking.

Rank	Contestant	Score
1	A	100
2	B	90
2	C	90
4	D	88

Looking at only the ranking column, we can see that this column will be a non-decreasing sequence of n integers and has some interesting attributes. Your task is to count the number of possible ranking sequences. Please note that two sequences A and B are considered different iff there exists a position i where $A_i \neq B_i$.

Input

The input contains a single integer n ($1 \leq n \leq 50$).

Output

The output should contain the number of possible ranking sequences.

Explanation of the sample

The ranking of 3 students could be one of the following scenarios:

- `1 1 1` Three students have the same score;
- `1 1 3` Two students tied for the highest score and they are better than the third student;
- `1 2 2` One student has the highest score, the other two have the same score;
- `1 2 3` Three students have different scores.

Sample Input 1	Sample Output 1
3	4

Problem G
Group Testing

During the first outbreak of the COVID-19 pandemic, extensive population-wide testing proved to be one of the best strategy to control and prevent the outbreak. However, PCR testing costs both time and money. An effective way to speed up the process and to save our resources is to implement group testing (or pool testing) strategy.

In this method, instead of doing test for individuals, we mix the samples of p individuals and run the test. If the result is negative, we can safely conclude that all p individuals are negative. Otherwise, if the result is positive, individual testing is used to determine who has the virus.

In an area, there are n families indexed from 0 to $n-1$. The i-th family has a_i members. Each family will form a group. After running n tests for n groups, we have exactly k groups with positive results.

Your task is to calculate the minimum and maximum number of extra individual tests we have to do.

Input

- The first line contains 2 integers n and k. $(0 \le k \le n \le 1\,000)$
- The second line contains n integers $a_0, a_1, \cdots, a_{n-1}$. $(1 \le a_i \le 10)$

Output

You should print 2 integers, the minimum and maximum number of extra individual tests we have to do.

Explanation of the samples

- In the first sample, if the positive groups are 0 and 1, we have to do 5 more tests; if the positive group are 3 and 4, we have to do 11 more tests;
- In the second sample, if the positive group is 0, we don't need any individual tests because this group has only 1 member.

Sample Input 1

```
5 2
2 3 4 5 6
```

Sample Output 1

```
5 11
```

Sample Input 2

```
3 1
1 2 3
```

Sample Output 2

```
0 3
```

Problem H
Hallway Tiling

Last year, the campus of FPT University was used as an isolation facility for COVID-19 patients. Now, the daily new cases have dropped significantly and the campus has been transformed back to a normal university environment. The university decided to re-tile their hallway.

The hallway of FPT university is a rectangle shape of size $r \cdot n$, this hallway will be re-tiled using $1 \cdot 2$ ceramic tiles. In this hallway, there are k positions which are pillars so we will not re-tile these positions.

Your task is to count the number of possible ways to re-tile the hallway.

Input

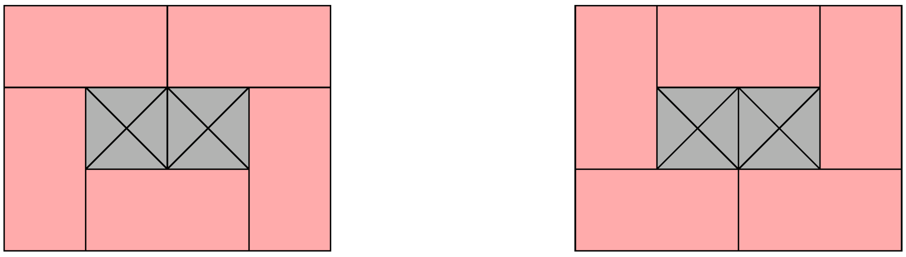
- The first line of the input consists of 3 integers r, n and k ($1 \le r \le 6, 1 \le n \le 10^{12}, 0 \le k \le 10$).
- Then k lines follows. Each contains 2 integers x and y describing the position of a pillar ($1 \le x \le r, 1 \le y \le n$). No 2 pillars are in the same position.

Output

Print a single integer denoting the number of possible ways to re-tile this hallway. Since this number could be large, you should print it modulo $10^9 + 7$.

Explanation of the sample

There are 2 ways to re-tile the hallway:



Sample Input 1

```
3 4 2
2 2
2 3
```

Sample Output 1

```
2
```

Problem I
ICPC Problem Selection

When it comes to organizing an ICPC, one of the most challenging job of the scientific comittee is to create a good problem set. Luckily, this year we received n task proposals. Each task proposal is tagged with at least one of the following categories: **dynamic programming** (dp), **graph theory** (graph), **math and geometry** (mathgeo), **data structure** (ds) and **adhoc** (adhoc).

The scientific commitee wants a balanced problem set. Thus, they introduced several selection rules:

- The number of problems with tag **dynamic programming** must be in range $[dp_{min}, dp_{max}]$.
- The number of problems with tag **graph theory** must be in range $[graph_{min}, graph_{max}]$.
- The number of problems with tag **math and geometry** must be in range $[mathgeo_{min}, mathgeo_{max}]$.
- The number of problems with tag **data structure** must be in range $[ds_{min}, ds_{max}]$.
- The number of problems with tag **adhoc** must be in range $[adhoc_{min}, adhoc_{max}]$.

Given the number of proposals n, the tags of every proposal and these numbers dp_{min}, dp_{max}, $graph_{min}$, $graph_{max}$, $mathgeo_{min}$, $mathgeo_{max}$, ds_{min}, ds_{max}, $adhoc_{min}$, $adhoc_{max}$; your task is to calculate the number of non-empty problem sets which satisfy all the above rules. Each problem set is a subset of the n given proposals. Two problem sets A and B are considered different iff there exists at least one proposal which is included in A but not in B, or vice versa.

Input

- The first line contains an integer n $(1 \le n \le 50)$ — the number of task proposals .
- In the next n lines, the i-th one describes the i-th proposal. It starts with an integer t_i $(1 \le t_i \le 5)$ denoting the number of tags assigned to the i-th proposal, and follows by t_i pair-wise distinct strings describing these tags. Each string is one of the categories listed above.
- The next line contains 2 integers dp_{min} and dp_{max} $(0 \le dp_{min} \le dp_{max} \le 15)$.
- The next line contains 2 integers $graph_{min}$ and $graph_{max}$ $(0 \le graph_{min} \le graph_{max} \le 15)$.
- The next line contains 2 integers $mathgeo_{min}$ and $mathgeo_{max}$ $(0 \le mathgeo_{min} \le mathgeo_{max} \le 15)$.
- The next line contains 2 integers ds_{min} and ds_{max} $(0 \le ds_{min} \le ds_{max} \le 15)$.
- The last line contains 2 integers $adhoc_{min}$ and $adhoc_{max}$ $(0 \le adhoc_{min} \le adhoc_{max} \le 15)$.

Output

Print a single integer denoting the number of different problem sets that satisfy the selection rules.

Explanation of the samples

- In the first sample, we must use all proposals 3, 4, and 5. Among proposals 1, 2, and 6; there are 5 ways to use some of them: $\{1, 2\}$, $\{1, 6\}$, $\{2, 6\}$, $\{1, 2, 6\}$ and $\{6\}$.
- In the second sample, we don't have any proposals with the tag **math and geometry**, thus there is no satisfying problem set.

Sample Input 1

```
6
1 dp
1 graph
1 mathgeo
1 ds
1 adhoc
2 dp graph
1 2
1 2
1 2
1 2
1 2
```

Sample Output 1

```
5
```

Sample Input 2

```
2
2 dp adhoc
2 ds graph
1 2
1 2
1 2
1 2
1 2
```

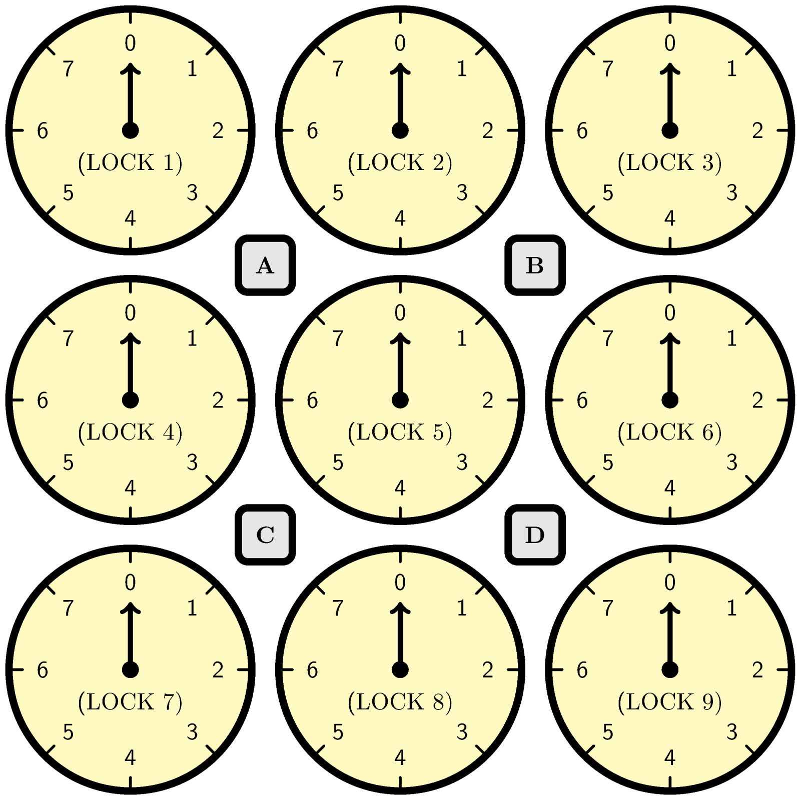
Sample Output 2

```
0
```

Problem J
Jewel Box

Minh has found an old mystery jewel box. On the top side of the box, there is a puzzle. It seems that solving this puzzle will open this box.

The puzzle has 9 circular locks and 4 pins. The locks are numbered from 1 to 9, the pins are called A, B, C and D. They are positioned like the following image:

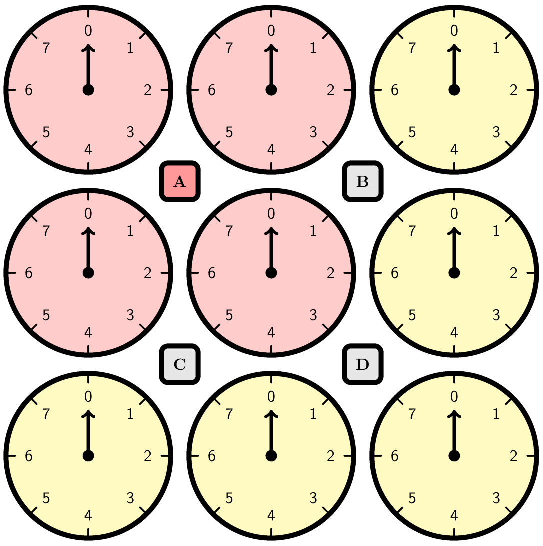


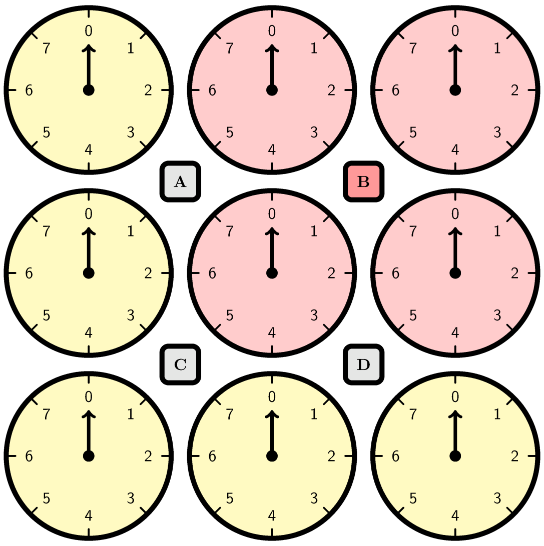
After playing with the jewel box for a while, Minh realized some properties of this box:

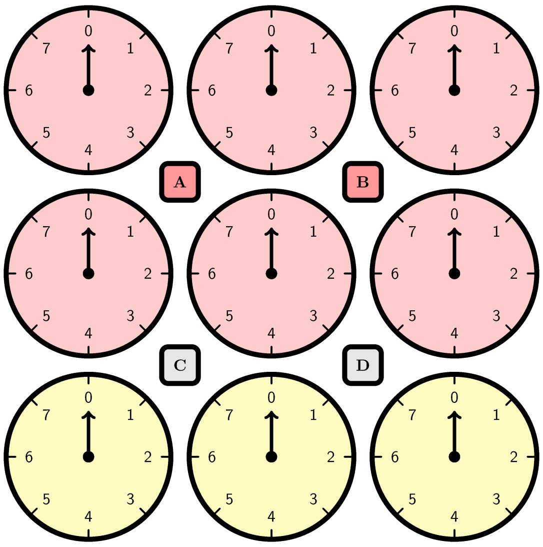
- Each lock has C numbers from 0 to $C - 1$.
- You only can rotate the 4 locks in the corners $1, 3, 7$ and 9; and only in clockwise direction.
- When a lock is rotated, if the current value of the lock is x, the new value of the lock will be $(x + 1)$ modulo C.
- You can turn on or turn off the pins as many times as you want.
- When a pin is turned on, 4 locks around that pin will rotate synchronously. In other words, if one lock is rotating, the other 3 locks will also rotate in the same way. For example:
  - If pin A is turned on, rotating lock 1 will cause locks $2, 4, 5$ to rotate.

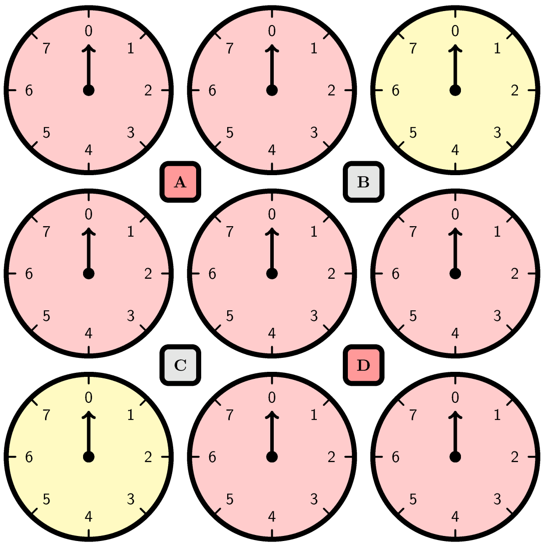
- If pin B is turned on, rotating lock 3 will cause locks $2, 5, 6$ to rotate.
- If pins A and B are turned on while pins C and D are turned off, rotating lock 1 will cause locks $2, 4, 5$ to rotate. Because pin B is turned on, locks 2 and 5 rotating will also cause locks 3 and 6 to rotate. Thus, with pin A and pin B turned on, rotating lock 1 will cause 6 locks to rotate.
- If pin A and pin D are both turned on while pins B and C are turned off, locks $1, 2, 4, 5, 6, 8, 9$ will be synchronous.
- If pin A is turned on and all other pins are turned off, rotating lock 9 will only affect itself.

The following figures show the synchronous locks when some pins are turned on. Red and grey represent the on and off states, respectively.









Given the current state of the locks, your task is to help Minh to make all the locks to value 0. Please note that, you don't have to minimize the number of moves.

Input

The input starts with an integer C ($8 \leq C \leq 50$). Then, exactly 3 lines follow, each line contains exactly 3 integers between 0 and $C - 1$, inclusive, representing the initial state of the 9 locks.

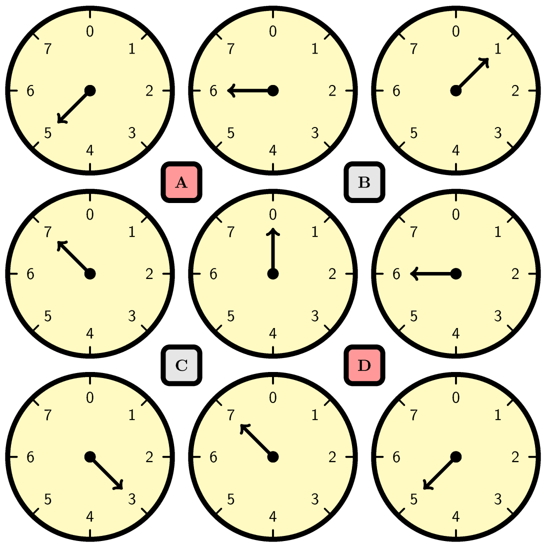
Output

If there is no solution, print a single line containing `NO`. Otherwise, print `YES` on the first line, followed by:

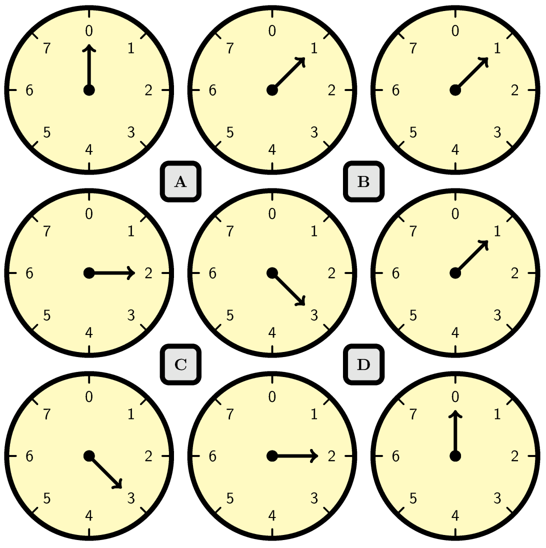
- A single line containing k — the number of steps ($0 \leq k \leq 1\,000$).
- In the next k lines, each contains 6 integers $p_A, p_B, p_C, p_D, c, rot$, where:
  - $p_A, p_B, p_C, p_D \in \{0, 1\}$ — represent the state of the pins: 1 represents the on state, while 0 represents the off state.
  - $c \in \{1, 3, 7, 9\}$ — the lock that we rotate.
  - $1 \leq rot \leq 100$ — the number of times we rotate.

Sample clarification

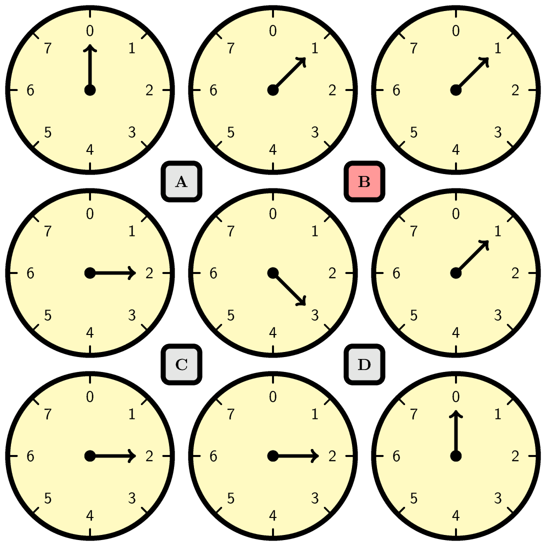
There are multiple solutions for this test case, this is one of them:

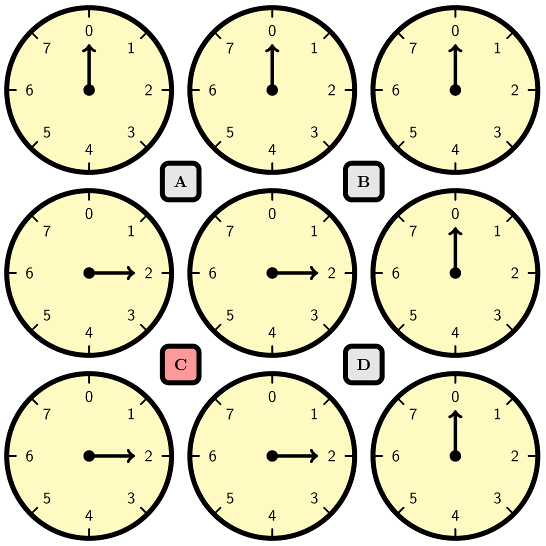


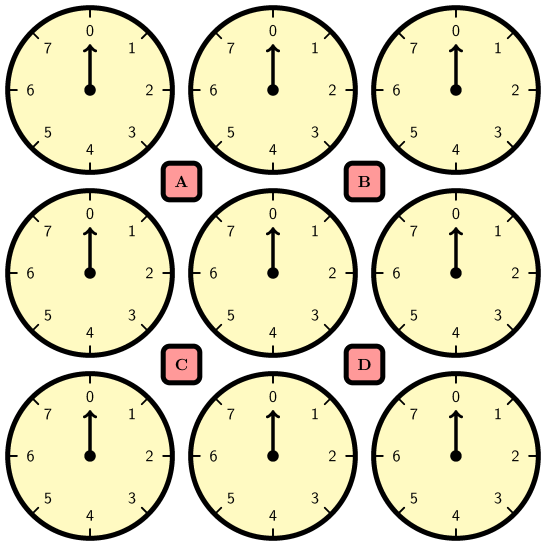
This is the initial state of the locks. With pin A and pin D on, we rotate lock 1 3 times.



Turn off pin A and D. With all pins turned off, we rotate lock 7 7 times.


Now, we turn on pin B, and rotate lock 3 7 times.


One last move, turn off pin B, turn on pin C, and rotate lock 7 6 times.


We got what we want!

Sample Input 1

```
8
5 6 1
7 0 6
3 7 5
```

Sample Output 1

```
YES
4
1 0 0 1 1 3
0 0 0 0 7 7
0 1 0 0 3 7
0 0 1 0 7 6
```

Problem K
K-query

You are given an array of balls. Each ball is either black or white. Initially, the array has n balls. You are about to proceed q queries one by one. In each query, let m be the number of balls in the array before the query happens and number the balls from 1 to m (inclusive). Each query is in one of the following forms:

- `I x k c` ($0 \le x \le m$, $1 \le k \le 10^9$, c is `B` or `W`): insert k balls of color c to the array, right after the x-th ball. More precisely:
  - If $x = 0$, balls are inserted at the beginning of the array.
  - If $1 \le x < m$, balls are inserted between the x-th and the $x + 1$-th ball of the array.
  - If $x = m$, balls are inserted at the end of the array.
  - If c equals `B`, all inserted balls are black.
  - If c equals `W`, all inserted balls are white.
- `D x k` ($1 \le x \le m$, $1 \le k \le m - x + 1$): Delete k consecutive balls from the array, starting at the x-th one.
- `F x k` ($1 \le x \le m$, $1 \le k \le m - x + 1$): Flip the color of k consecutive balls of the array starting at the x-th one. In other words, replace every black ball with a white one at the same position, and vice versa.
- `Q x k` ($1 \le x \le m$, $1 \le k \le m - x + 1$): Consider only k consecutive balls of the array starting at the x-th one, divide them into a minimum number of segments so that every segment contains consecutive balls of the same color, and let $l_1, l_2, \ldots, l_t$ be the lengths of these segments. Compute and print the value $l_1^2 + l_2^2 + \ldots + l_t^2$. Note that the way to divide the balls is always unique.

Input

- The first line contains two integers n and q ($1 \le n, q \le 200\,000$).
- The second line contains n characters `B` and `W`, denoting the color of the balls in the initial array.
- For the last q lines, each line contains a query as described above.

Output

For each query of the form `Q x k`, output its result in a line. As the result can be rather large, output it modulo $998\,244\,353$.

Explanation of the samples

For the first sample, the array is static:

- In the first query, the considered balls are `WW`. There is only one segment, the result is $2^2 = 4$.
- In the second query, the considered balls are `WWBBWW`. We devide these balls into three segments: 2 white balls, 2 black balls and 2 white balls. Hence, the result is $2^2+2^2+2^2 = 12$.

For the second sample:

- Initially, the array is `BBWBB`.
- In the first query, the considered balls are `BWB`. We divide these balls into three segments of one ball each. Hence, the result is $1^2 + 1^2 + 1^2 = 3$.
- After the second query, the array is `BBBBBWBB`.
- In the third query, the considered balls are `BBBW`. We divide these balls into two segments: the first one contains 3 black balls and the second one contains one white ball. Hence, the result is $3^2 + 1^2 = 10$.
- After the forth query, the array is `BBBWWBBB`.
- In the fifth query, the considered balls are `WW`. Since all balls are white, we can put them into one segment. Hence, the result is $2^2 = 4$.
- After the sixth query, the array is `BBBBBB`.
- In the seventh query, we consider the whole array. Since all balls are black, we can put them into one segment. Hence, the result is $6^2 = 36$.

Sample Input 1

```
8 2
WWWBBWWW
Q 1 2
Q 2 6
```

Sample Output 1

```
4
12
```

Sample Input 2

```
5 7
BBWBB
Q 2 3
I 0 3 B
Q 3 4
F 4 3
Q 4 2
D 4 2
Q 1 6
```

Sample Output 2

```
3
10
4
36
```

Problem L
Lucky Pair

A pair of positive integers (x, y) is considered a *lucky pair* iff there exists a positive integer k such that: $x^k + y^k$ is divisible by $x \cdot y$. For example:

- $(2, 4)$ is a lucky pair because $2^3 + 4^3 = 8 + 64 = 72$ is divisible by $2 \cdot 4 = 8$;
- $(3, 3)$ is a lucky pair because $3^2 + 3^2 = 9 + 9 = 18$ is divisible by $3 \cdot 3 = 9$;
- $(1, 2)$ is not a lucky pair because $1^k + 2^k$ is always odd for every $k > 0$ and can not be divisible by $1 \cdot 2 = 2$.

You are given an array a containing n positive integers, your task is to count the number of pairs (i, j) so that $i < j$ and (a_i, a_j) is a lucky pair.

Input

- The first line contains a single integer n — the length of the array $(2 \leq n \leq 3 \cdot 10^5)$.
- The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10^7)$ — the elements of the array.

Output

Write a single integer denoting the number of lucky pairs in the array a.

Sample Input 1	Sample Output 1
6 1 2 3 4 5 6	1

Sample Input 2	Sample Output 2
4 7 7 7 7	6

Problem M
Master Chef

Hoang Yen has recently won the television show Master Chef Vietnam. She decided to throw a big party to celebrate the title with her friends.

Hoang Yen will prepare her wonderful recipes all by herself. In order to show as many recipes as possible, she decided:

- Each table will be served a menu of exactly n dishes.
- Each dish will not appear more than n times.
- For each pair of tables, there is **exactly** 1 common dish.

Given an integer n where $n - 1$ has at most 2 positive divisors, what is the maximal number of tables she can serve?

Input

The input consists of a single integer n ($1 \le n \le 100$). It is guaranteed that $n - 1$ has at most 2 positive divisors.

Output

- The first line contains an integer t — the maximal number of tables.
- In the next t lines, the i-th one contains n distinct integers $d_{i,1}, d_{i,2}, \cdots, d_{i,n}$ describing n dishes for table i ($1 \le d_{i,j}$). Dishes should be numbered consecutively from 1.

Sample Input 1	Sample Output 1

2	3
	1 2
	2 3
	3 1