Codeforces Round #738 (Div. 2)

A. Mocha and Math

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mocha is a young girl from high school. She has learned so much interesting knowledge from her teachers, especially her math teacher. Recently, Mocha is learning about binary system and very interested in bitwise operation.

This day, Mocha got a sequence a of length n. In each operation, she can select an arbitrary interval $[l, r]$ and for all values i ($0 \leq i \leq r - l$), replace a_{l+i} with $a_{l+i} \,\&\, a_{r-i}$ at the same time, where $\&$ denotes the bitwise AND operation. This operation can be performed **any number of times**.

For example, if $n = 5$, the array is $[a_1, a_2, a_3, a_4, a_5]$, and Mocha selects the interval $[2, 5]$, then the new array is $[a_1, a_2 \,\&\, a_5, a_3 \,\&\, a_4, a_4 \,\&\, a_3, a_5 \,\&\, a_2]$.

Now Mocha wants to minimize the maximum value in the sequence. As her best friend, can you help her to get the answer?

Input
Each test contains multiple test cases.

The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases. Each test case consists of two lines.

The first line of each test case contains a single integer n ($1 \leq n \leq 100$) — the length of the sequence.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leq a_i \leq 10^9$).

Output
For each test case, print one integer — the minimal value of the maximum value in the sequence.

Example

input
4 2 1 2 3 1 1 3 4 3 11 3 7 5 11 7 15 3 7
output
0 1 3 3

Note
In the first test case, Mocha can choose the interval $[1, 2]$, then the sequence becomes $[0, 0]$, where the first element is $1 \,\&\, 2$, and the second element is $2 \,\&\, 1$.

In the second test case, Mocha can choose the interval $[1, 3]$, then the sequence becomes $[1, 1, 1]$, where the first element is $1 \,\&\, 3$, the second element is $1 \,\&\, 1$, and the third element is $3 \,\&\, 1$.

B. Mocha and Red and Blue

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

As their story unravels, a timeless tale is told once again...

Shirahime, a friend of Mocha's, is keen on playing the music game Arcaea and sharing Mocha interesting puzzles to solve. This day, Shirahime comes up with a new simple puzzle and wants Mocha to solve them. However, these puzzles are too easy for Mocha to solve, so she wants you to solve them and tell her the answers. The puzzles are described as follow.

There are n squares arranged in a row, and each of them can be painted either red or blue.

Among these squares, some of them have been painted already, and the others are blank. You can decide which color to paint on each blank square.

Some pairs of adjacent squares may have the same color, which is imperfect. We define the *imperfectness* as the number of pairs of adjacent squares that share the same color.

For example, the imperfectness of "BRRRBBR" is 3, with "BB" occurred once and "RR" occurred twice.

Your goal is to minimize the imperfectness and print out the colors of the squares after painting.

Input
Each test contains multiple test cases.

The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases. Each test case consists of two lines.

The first line of each test case contains an integer n ($1 \le n \le 100$) — the length of the squares row.

The second line of each test case contains a string s with length n, containing characters 'B', 'R' and '?'. Here 'B' stands for a blue square, 'R' for a red square, and '?' for a blank square.

Output
For each test case, print a line with a string only containing 'B' and 'R', the colors of the squares after painting, which imperfectness is minimized. If there are multiple solutions, print any of them.

Example

input
5 7 ?R???BR 7 ???R??? 1 ? 1 B 10 ?R??RB??B?
output
BRRBRBR BRBRBRB B B BRRBRBBRBR

Note
In the first test case, if the squares are painted "BRRBRBR", the imperfectness is 1 (since squares 2 and 3 have the same color), which is the minimum possible imperfectness.

C. Mocha and Hiking

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The city where Mocha lives in is called Zhijiang. There are $n+1$ villages and $2n-1$ directed roads in this city.

There are two kinds of roads:

- $n-1$ roads are from village i to village $i+1$, for all $1 \le i \le n-1$.
- n roads can be described by a sequence $a_1, \ldots, a_n$. If $a_i = 0$, the i-th of these roads goes from village i to village $n+1$, otherwise it goes from village $n+1$ to village i, for all $1 \le i \le n$.

Mocha plans to go hiking with Taki this weekend. To avoid the trip being boring, they plan to go through every village **exactly once**. They can start and finish at any villages. Can you help them to draw up a plan?

Input
Each test contains multiple test cases.

The first line contains a single integer t ($1 \le t \le 20$) — the number of test cases. Each test case consists of two lines.

The first line of each test case contains a single integer n ($1 \le n \le 10^4$) — indicates that the number of villages is $n+1$.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 1$). If $a_i = 0$, it means that there is a road from village i to village $n+1$. If $a_i = 1$, it means that there is a road from village $n+1$ to village i.

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output
For each test case, print a line with $n+1$ integers, where the i-th number is the i-th village they will go through. If the answer doesn't exist, print -1.

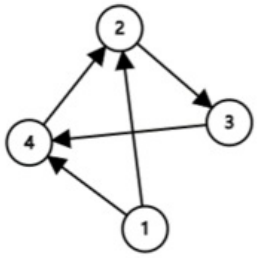
If there are multiple correct answers, you can print any one of them.

Example

input
2 3 0 1 0 3 1 1 0
output
1 4 2 3 4 1 2 3

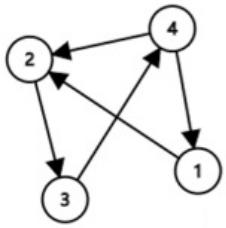
Note

In the first test case, the city looks like the following graph:



So all possible answers are $(1 \to 4 \to 2 \to 3)$, $(1 \to 2 \to 3 \to 4)$.

In the second test case, the city looks like the following graph:



So all possible answers are $(4 \to 1 \to 2 \to 3)$, $(1 \to 2 \to 3 \to 4)$, $(3 \to 4 \to 1 \to 2)$, $(2 \to 3 \to 4 \to 1)$.

D1. Mocha and Diana (Easy Version)

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is the easy version of the problem. The only difference between the two versions is the constraint on n. You can make hacks only if all versions of the problem are solved.

A forest is an undirected graph without cycles (not necessarily connected).

Mocha and Diana are friends in Zhijiang, both of them have a forest with nodes numbered from 1 to n, and they would like to add edges to their forests such that:

- After adding edges, both of their graphs are still forests.
- They add the same edges. That is, if an edge (u, v) is added to Mocha's forest, then an edge (u, v) is added to Diana's forest, and vice versa.

Mocha and Diana want to know the maximum number of edges they can add, and which edges to add.

Input

The first line contains three integers n, m_1 and m_2 ($1 \le n \le 1000$, $0 \le m_1, m_2 < n$) — the number of nodes and the number of initial edges in Mocha's forest and Diana's forest.

Each of the next m_1 lines contains two integers u and v ($1 \le u, v \le n$, $u \ne v$) — the edges in Mocha's forest.

Each of the next m_2 lines contains two integers u and v ($1 \le u, v \le n$, $u \ne v$) — the edges in Diana's forest.

Output

The first line contains only one integer h, the maximum number of edges Mocha and Diana can add (in each forest).

Each of the next h lines contains two integers u and v ($1 \le u, v \le n$, $u \ne v$) — the edge you add each time.

If there are multiple correct answers, you can print any one of them.

Examples

input
3 2 2 1 2 2 3 1 2 1 3
output
0

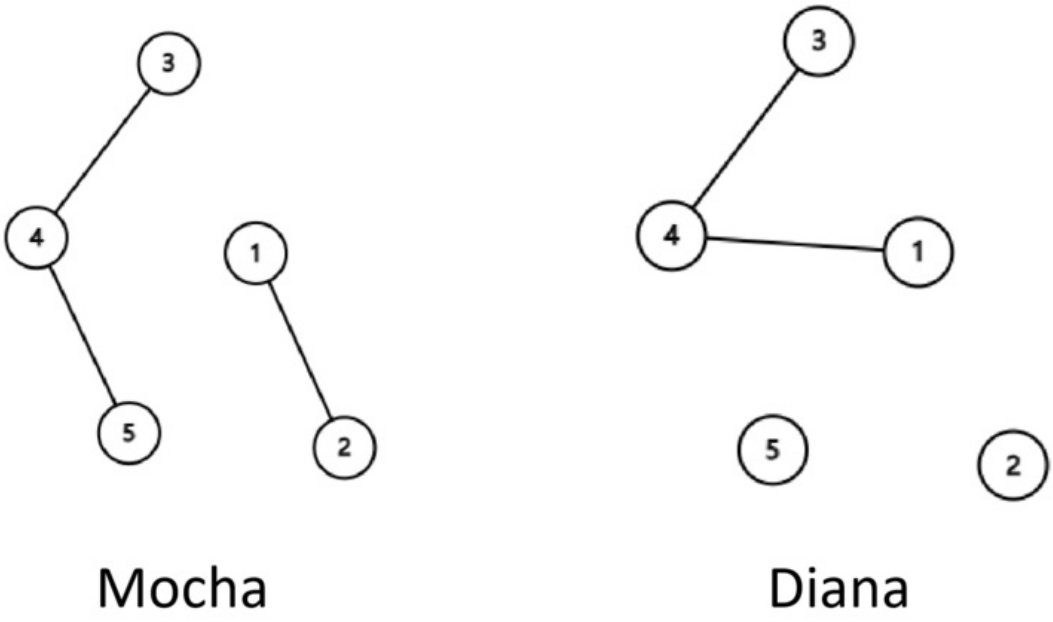
input
5 3 2 5 4 2 1 4 3 4 3 1 4
output
1 2 4

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

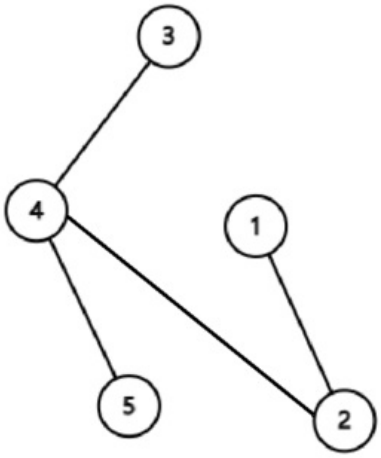
Note

In the first example, we cannot add any edge.

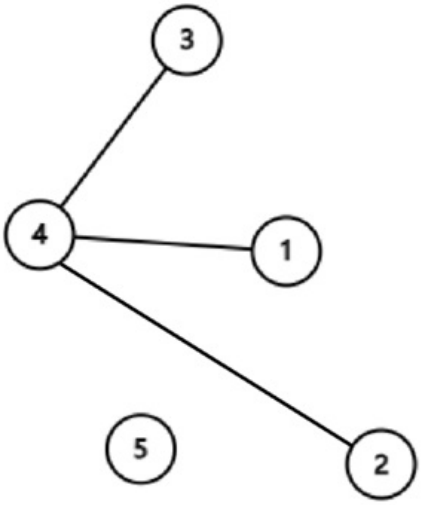
In the second example, the initial forests are as follows.



We can add an edge $(2, 4)$.



Mocha



Diana

D2. Mocha and Diana (Hard Version)

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is the hard version of the problem. The only difference between the two versions is the constraint on n. You can make hacks only if all versions of the problem are solved.

A forest is an undirected graph without cycles (not necessarily connected).

Mocha and Diana are friends in Zhijiang, both of them have a forest with nodes numbered from 1 to n, and they would like to add edges to their forests such that:

- After adding edges, both of their graphs are still forests.
- They add the same edges. That is, if an edge (u, v) is added to Mocha's forest, then an edge (u, v) is added to Diana's forest, and vice versa.

Mocha and Diana want to know the maximum number of edges they can add, and which edges to add.

Input

The first line contains three integers n, m_1 and m_2 ($1 \leq n \leq 10^5$, $0 \leq m_1, m_2 < n$) — the number of nodes and the number of initial edges in Mocha's forest and Diana's forest.

Each of the next m_1 lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the edges in Mocha's forest.

Each of the next m_2 lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the edges in Diana's forest.

Output

The first line contains only one integer h, the maximum number of edges Mocha and Diana can add.

Each of the next h lines contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$) — the edge you add each time.

If there are multiple correct answers, you can print any one of them.

Examples

input
3 2 2 1 2 2 3 1 2 1 3
output
0

input

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

output

```
1
2 4
```

input

```
8 1 2
1 7
2 6
1 5
```

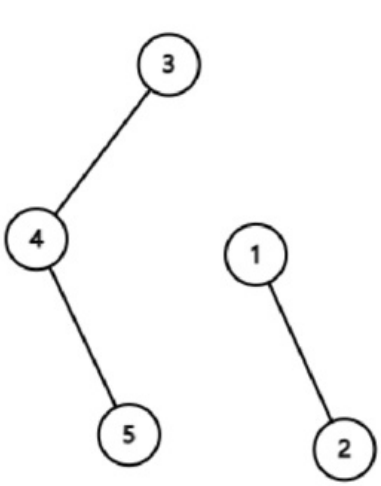
output

```
5
5 2
2 3
3 4
4 7
6 8
```

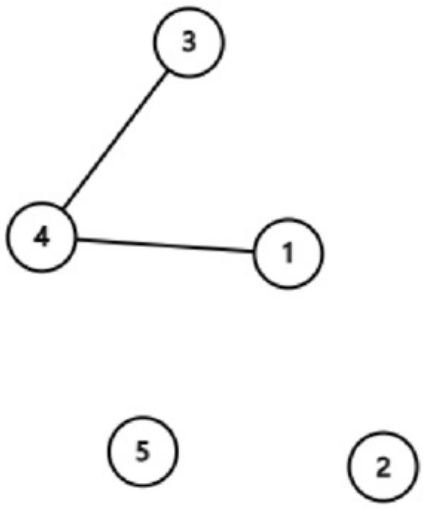
Note

In the first example, we cannot add any edge.

In the second example, the initial forests are as follows.

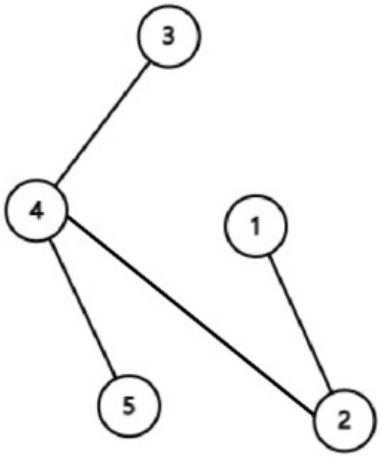


Mocha

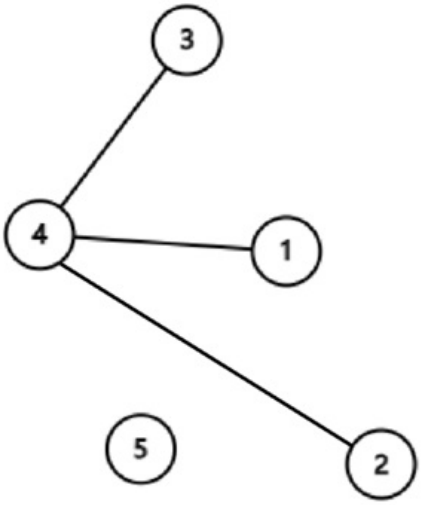


Diana

We can add an edge $(2, 4)$.



Mocha



Diana

E. Mocha and Stars

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mocha wants to be an astrologer. There are n stars which can be seen in Zhijiang, and the brightness of the i-th star is a_i.

Mocha considers that these n stars form a constellation, and she uses $(a_1, a_2, \ldots, a_n)$ to show its state. A state is called *mathematical* if all of the following three conditions are satisfied:

- For all i ($1 \leq i \leq n$), a_i is an integer in the range $[l_i, r_i]$.
- $\sum_{i=1}^{n} a_i \leq m$.
- $\gcd(a_1, a_2, \ldots, a_n) = 1$.

Here, $\gcd(a_1, a_2, \ldots, a_n)$ denotes the greatest common divisor (GCD) of integers $a_1, a_2, \ldots, a_n$.

Mocha is wondering how many different mathematical states of this constellation exist. Because the answer may be large, you must find it modulo $998\,244\,353$.

Two states $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ are considered different if there exists i ($1 \leq i \leq n$) such that $a_i \neq b_i$.

Input

The first line contains two integers n and m ($2 \leq n \leq 50$, $1 \leq m \leq 10^5$) — the number of stars and the upper bound of the sum of the brightness of stars.

Each of the next n lines contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq m$) — the range of the brightness of the i-th star.

Output

Print a single integer — the number of different mathematical states of this constellation, modulo $998\,244\,353$.

Examples

input
2 4 1 3 1 2
output
4

input
5 10 1 10 1 10 1 10 1 10 1 10

1 10

output

251

input

5 100
1 94
1 96
1 91
4 96
6 97

output

47464146

Note

In the first example, there are 4 different mathematical states of this constellation:

- $a_1 = 1$, $a_2 = 1$.
- $a_1 = 1$, $a_2 = 2$.
- $a_1 = 2$, $a_2 = 1$.
- $a_1 = 3$, $a_2 = 1$.