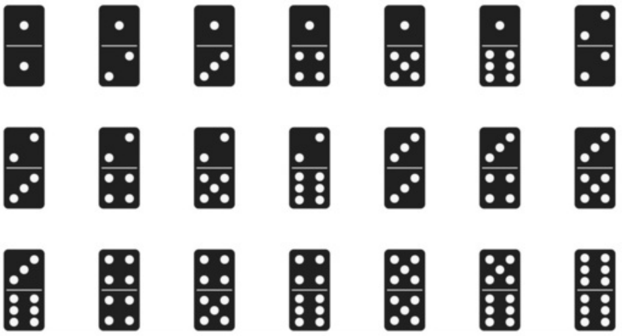
A. Anadi and Domino

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

Anadi has a set of dominoes. Every domino has two parts, and each part contains some dots. For every a and b such that $1 \leq a \leq b \leq 6$, there is exactly one domino with a dots on one half and b dots on the other half. The set contains exactly 21 dominoes. Here is an exact illustration of his set:



Also, Anadi has an undirected graph without self-loops and multiple edges. He wants to choose some dominoes and place them on the edges of this graph. He can use at most one domino of each type. Each edge can fit at most one domino. It's not necessary to place a domino on each edge of the graph.

When placing a domino on an edge, he also chooses its direction. In other words, one half of any placed domino must be directed toward one of the endpoints of the edge and the other half must be directed toward the other endpoint. There's a catch: if there are multiple halves of dominoes directed toward the same vertex, each of these halves must contain the same number of dots.

How many dominoes at most can Anadi place on the edges of his graph?

Input

The first line contains two integers n and m ($1 \leq n \leq 7$, $0 \leq m \leq \frac{n\cdot(n-1)}{2}$) — the number of vertices and the number of edges in the graph.

The next m lines contain two integers each. Integers in the i-th line are a_i and b_i ($1 \leq a, b \leq n$, $a \neq b$) and denote that there is an edge which connects vertices a_i and b_i.

The graph might be disconnected. It's however guaranteed that the graph doesn't contain any self-loops, and that there is at most one edge between any pair of vertices.

Output

Output one integer which denotes the maximum number of dominoes which Anadi can place on the edges of the graph.

Examples

input

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

output

```
4
```

input

```
7 0
```

output

```
0
```

input

```
3 1
1 3
```

output

```
1
```

input

```
7 21
1 2
1 3
1 4
1 5
1 6
1 7
2 3
2 4
2 5
2 6
2 7
3 4
3 5
3 6
3 7
4 5
```

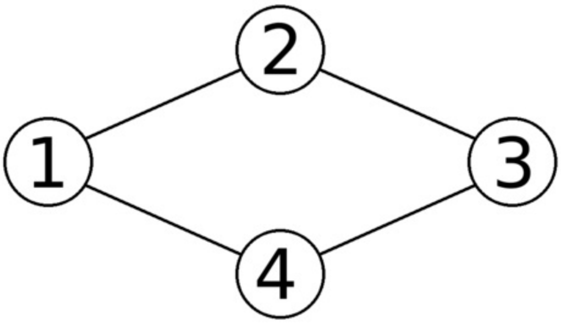
```
4 6
4 7
5 6
5 7
6 7
```

output

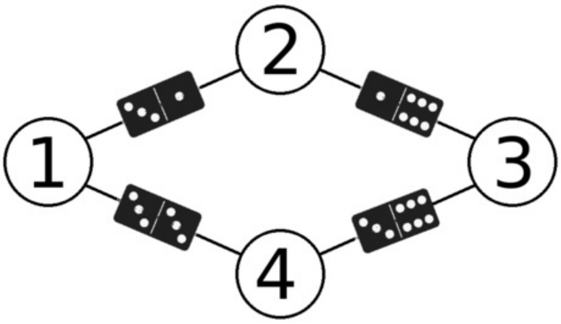
```
16
```

Note

Here is an illustration of Anadi's graph from the first sample test:



And here is one of the ways to place a domino on each of its edges:



Note that each vertex is faced by the halves of dominoes with the same number of dots. For instance, all halves directed toward vertex 1 have three dots.

B. Marcin and Training Camp

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Marcin is a coach in his university. There are n students who want to attend a training camp. Marcin is a smart coach, so he wants to send only the students that can work calmly with each other.

Let's focus on the students. They are indexed with integers from 1 to n. Each of them can be described with two integers a_i and b_i; b_i is equal to the skill level of the i-th student (the higher, the better). Also, there are 60 known algorithms, which are numbered with integers from 0 to 59. If the i-th student knows the j-th algorithm, then the j-th bit (2^j) is set in the binary representation of a_i. Otherwise, this bit is not set.

Student x thinks that he is better than student y if and only if x knows some algorithm which y doesn't know. Note that two students can think that they are better than each other. A group of students can work together calmly if no student in this group thinks that he is better than everyone else in this group.

Marcin wants to send a group of at least two students which will work together calmly and will have the maximum possible sum of the skill levels. What is this sum?

Input

The first line contains one integer n ($1 \le n \le 7000$) — the number of students interested in the camp.

The second line contains n integers. The i-th of them is a_i ($0 \le a_i < 2^{60}$).

The third line contains n integers. The i-th of them is b_i ($1 \le b_i \le 10^9$).

Output

Output one integer which denotes the maximum sum of b_i over the students in a group of students which can work together calmly. If no group of at least two students can work together calmly, print `0`.

Examples

input

```
4
3 2 3 6
2 8 5 10
```

output

```
15
```

input

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

output

```
0
```

input

```
1
0
1
```

output

```
0
```

Note

In the first sample test, it's optimal to send the first, the second and the third student to the camp. It's also possible to send only the first and the third student, but they'd have a lower sum of b_i.

In the second test, in each group of at least two students someone will always think that he is better than everyone else in the subset.

C. Kamil and Making a Stream

time limit per test: 4 seconds
memory limit per test: 768 megabytes
input: standard input
output: standard output

Kamil likes streaming the competitive programming videos. His MeTube channel has recently reached 100 million subscribers. In order to celebrate this, he posted a video with an interesting problem he couldn't solve yet. Can you help him?

You're given a tree — a connected undirected graph consisting of n vertices connected by $n-1$ edges. The tree is rooted at vertex 1. A vertex u is called an *ancestor* of v if it lies on the shortest path between the root and v. In particular, a vertex is an ancestor of itself.

Each vertex v is assigned its *beauty* x_v — a non-negative integer not larger than 10^{12}. This allows us to define the beauty of a path. Let u be an ancestor of v. Then we define the beauty $f(u, v)$ as the greatest common divisor of the beauties of all vertices on the shortest path between u and v. Formally, if $u = t_1, t_2, t_3, \dots, t_k = v$ are the vertices on the shortest path between u and v, then $f(u, v) = \gcd(x_{t_1}, x_{t_2}, \dots, x_{t_k})$. Here, $\gcd$ denotes the greatest common divisor of a set of numbers. In particular, $f(u, u) = \gcd(x_u) = x_u$.

Your task is to find the sum

$$\sum_{u \text{ is an ancestor of } v} f(u, v).$$

As the result might be too large, please output it modulo $10^9 + 7$.

Note that for each y, $\gcd(0, y) = \gcd(y, 0) = y$. In particular, $\gcd(0, 0) = 0$.

Input

The first line contains a single integer n ($2 \le n \le 100\,000$) — the number of vertices in the tree.

The following line contains n integers $x_1, x_2, \dots, x_n$ ($0 \le x_i \le 10^{12}$). The value x_v denotes the beauty of vertex v.

The following $n-1$ lines describe the edges of the tree. Each of them contains two integers a, b ($1 \le a, b \le n$, $a \ne b$) — the vertices connected by a single edge.

Output

Output the sum of the beauties on all paths (u, v) such that u is ancestor of v. This sum should be printed modulo $10^9 + 7$.

Examples

input

```
5
4 5 6 0 8
1 2
1 3
1 4
4 5
```

output

```
42
```

input

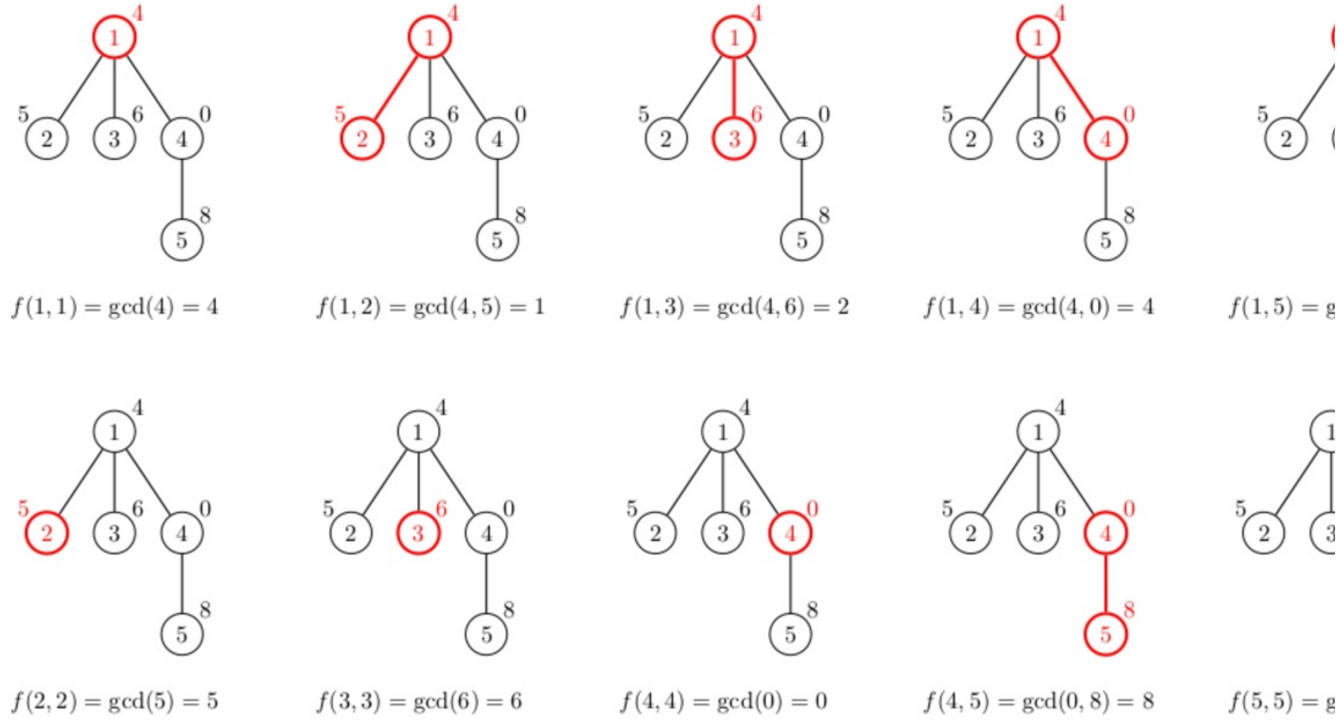
```
7
0 2 3 0 0 0 0
1 2
1 3
2 4
2 5
3 6
3 7
```

output

```
30
```

Note

The following figure shows all 10 possible paths for which one endpoint is an ancestor of another endpoint. The sum of beauties of all these paths is equal to 42:



D. Konrad and Company Evaluation

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Konrad is a Human Relations consultant working for VoltModder, a large electrical equipment producer. Today, he has been tasked with evaluating the level of happiness in the company.

There are n people working for VoltModder, numbered from 1 to n. Each employee earns a different amount of money in the company — initially, the i-th person earns i rubles per day.

On each of q following days, the salaries will be revised. At the end of the i-th day, employee v_i will start earning $n+i$ rubles per day and will become the best-paid person in the company. The employee will keep his new salary until it gets revised again.

Some pairs of people don't like each other. This creates a great psychological danger in the company. Formally, if two people a and b dislike each other and a earns more money than b, employee a will brag about this to b. A *dangerous triple* is a triple of three employees a, b and c, such that a brags to b, who in turn brags to c. If a dislikes b, then b dislikes a.

At the beginning of each day, Konrad needs to evaluate the number of *dangerous triples* in the company. Can you help him do it?

Input

The first line contains two integers n and m ($1 \le n \le 100\,000$, $0 \le m \le 100\,000$) — the number of employees in the company and the number of pairs of people who don't like each other. Each of the following m lines contains two integers a_i, b_i ($1 \le a_i, b_i \le n$, $a_i \neq b_i$) denoting that employees a_i and b_i hate each other (that is, a_i dislikes b_i and b_i dislikes a_i). Each such relationship will be mentioned exactly once.

The next line contains an integer q ($0 \le q \le 100\,000$) — the number of salary revisions. The i-th of the following q lines contains a single integer v_i ($1 \le v_i \le n$) denoting that at the end of the i-th day, employee v_i will earn the most.

Output

Output $q+1$ integers. The i-th of them should contain the number of *dangerous triples* in the company at the beginning of the i-th day.

Examples

input

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

output

```
4
3
2
```

input

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

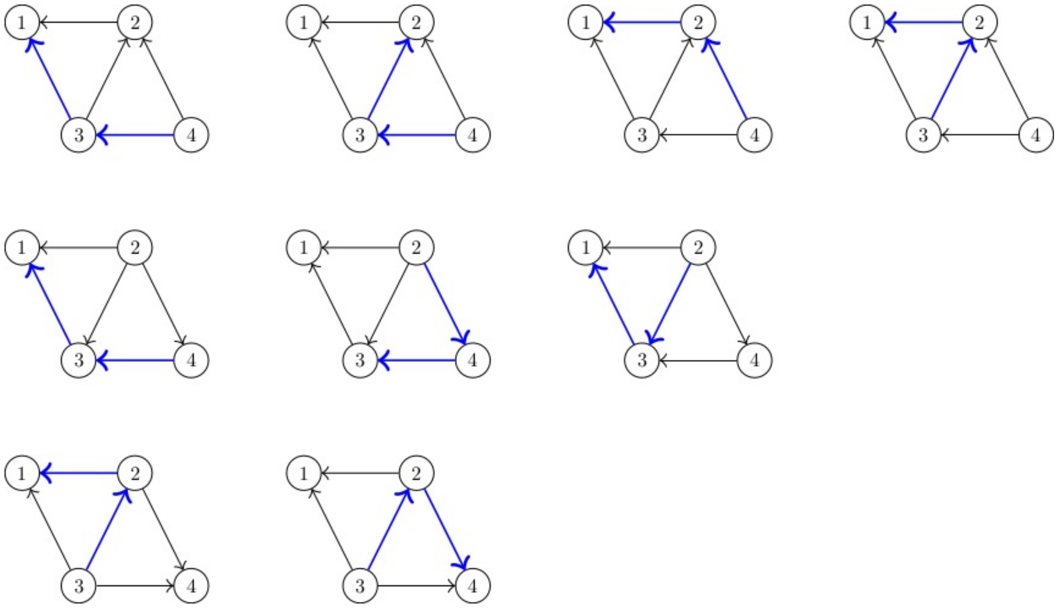
output

```
1
```

```
1
1
1
1
1
```

Note

Consider the first sample test. The i-th row in the following image shows the structure of the company at the beginning of the i-th day. A directed edge from a to b denotes that employee a brags to employee b. The dangerous triples are marked by highlighted edges.



E. Wojtek and Card Tricks

time limit per test: 3.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Wojtek has just won a maths competition in Byteland! The prize is admirable — a great book called 'Card Tricks for Everyone.' 'Great!' he thought, 'I can finally use this old, dusted deck of cards that's always been lying unused on my desk!'

The first chapter of the book is 'How to Shuffle k Cards in Any Order You Want.' It's basically a list of n intricate methods of shuffling the deck of k cards in a deterministic way. Specifically, the i-th recipe can be described as a permutation $(P_{i,1}, P_{i,2}, \dots, P_{i,k})$ of integers from 1 to k. If we enumerate the cards in the deck from 1 to k from top to bottom, then $P_{i,j}$ indicates the number of the j-th card from the top of the deck after the shuffle.

The day is short and Wojtek wants to learn only some of the tricks today. He will pick two integers l, r ($1 \le l \le r \le n$), and he will memorize each trick from the l-th to the r-th, inclusive. He will then take a sorted deck of k cards and repeatedly apply random memorized tricks until he gets bored. He still likes maths, so he started wondering: how many different decks can he have after he stops shuffling it?

Wojtek still didn't choose the integers l and r, but he is still curious. Therefore, he defined $f(l, r)$ as the number of different decks he can get if he memorizes all the tricks between the l-th and the r-th, inclusive. What is the value of

$$\sum_{l=1}^{n} \sum_{r=l}^{n} f(l, r)?$$

Input

The first line contains two integers n, k ($1 \le n \le 200\,000$, $1 \le k \le 5$) — the number of tricks and the number of cards in Wojtek's deck.

Each of the following n lines describes a single trick and is described by k distinct integers $P_{i,1}, P_{i,2}, \dots, P_{i,k}$ ($1 \le P_{i,j} \le k$).

Output

Output the value of the sum described in the statement.

Examples

input

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

output

```
25
```

input

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

output

```
31
```

Note

Consider the first sample:

- The first trick swaps two top cards.
- The second trick takes a card from the bottom and puts it on the top of the deck.

- The third trick swaps two bottom cards.

The first or the third trick allow Wojtek to generate only two distinct decks (either the two cards are swapped or not). Therefore, $f(1,1) = f(3,3) = 2$.

The second trick allows him to shuffle the deck in a cyclic order. Therefore, $f(2,2) = 3$.

It turns that two first tricks or two last tricks are enough to shuffle the deck in any way desired by Wojtek. Therefore, $f(1,2) = f(2,3) = f(1,3) = 3! = 6$.

F1. Marek and Matching (easy version)

time limit per test: 7 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is an easier version of the problem. In this version, $n \leq 6$.

Marek is working hard on creating strong testcases to his new algorithmic problem. You want to know what it is? Nah, we're not telling you. However, we can tell you how he generates the testcases.

Marek chooses an integer n and n^2 integers p_{ij} ($1 \leq i \leq n$, $1 \leq j \leq n$). He then generates a random bipartite graph with $2n$ vertices. There are n vertices on the left side: $\ell_1, \ell_2, \dots, \ell_n$, and n vertices on the right side: $r_1, r_2, \dots, r_n$. For each i and j, he puts an edge between vertices ℓ_i and r_j with probability p_{ij} percent.

It turns out that the tests will be strong only if a perfect matching exists in the generated graph. What is the probability that this will occur?

It can be shown that this value can be represented as $\frac{P}{Q}$ where P and Q are coprime integers and $Q \not\equiv 0 \pmod{10^9+7}$. Let Q^{-1} be an integer for which $Q \cdot Q^{-1} \equiv 1 \pmod{10^9+7}$. Print the value of $P \cdot Q^{-1}$ modulo 10^9+7.

Input

The first line of the input contains a single integer n ($\mathbf{1 \leq n \leq 6}$). The following n lines describe the probabilities of each edge appearing in the graph. The i-th of the lines contains n integers $p_{i1}, p_{i2}, \dots, p_{in}$ ($0 \leq p_{ij} \leq 100$); p_{ij} denotes the probability, in percent, of an edge appearing between ℓ_i and r_j.

Output

Print a single integer — the probability that the perfect matching exists in the bipartite graph, written as $P \cdot Q^{-1} \pmod{10^9+7}$ for P, Q defined above.

Examples

input

```
2
50 50
50 50
```

output

```
937500007
```

input

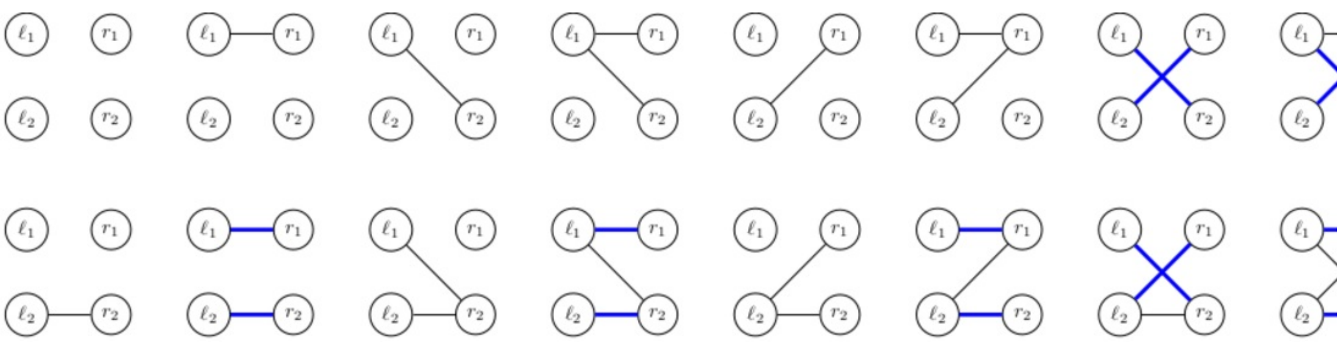
```
3
3 1 4
1 5 9
2 6 5
```

output

```
351284554
```

Note

In the first sample test, each of the 16 graphs below is equally probable. Out of these, 7 have a perfect matching:



Therefore, the probability is equal to $\frac{7}{16}$. As $16 \cdot 562\,500\,004 = 1 \pmod{10^9+7}$, the answer to the testcase is $7 \cdot 562\,500\,004 \mod (10^9+7) = 937\,500\,007$.

F2. Marek and Matching (hard version)

time limit per test: 15 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

This is a harder version of the problem. In this version, $n \leq 7$.

Marek is working hard on creating strong test cases to his new algorithmic problem. Do you want to know what it is? Nah, we're not telling you. However, we can tell you how he generates test cases.

Marek chooses an integer n and n^2 integers p_{ij} ($1 \leq i \leq n$, $1 \leq j \leq n$). He then generates a random bipartite graph with $2n$ vertices. There are n vertices on the left side: $\ell_1, \ell_2, \dots, \ell_n$, and n vertices on the right side: $r_1, r_2, \dots, r_n$. For each i and j, he puts an edge between vertices ℓ_i and r_j with probability p_{ij} percent.

It turns out that the tests will be strong only if a perfect matching exists in the generated graph. What is the probability that this will occur?

It can be shown that this value can be represented as $\frac{P}{Q}$ where P and Q are coprime integers and $Q \not\equiv 0 \pmod{10^9+7}$. Let Q^{-1} be an integer for which $Q \cdot Q^{-1} \equiv 1 \pmod{10^9+7}$. Print the value of $P \cdot Q^{-1}$ modulo 10^9+7.

Input

The first line of the input contains a single integer n ($\mathbf{1 \leq n \leq 7}$). The following n lines describe the probabilities of each edge appearing in the graph. The i-th of the lines contains n integers $p_{i1}, p_{i2}, \dots, p_{in}$ ($0 \leq p_{ij} \leq 100$); p_{ij} denotes the probability, in percent, of an edge appearing between ℓ_i and r_j.

Output

Print a single integer — the probability that the perfect matching exists in the bipartite graph, written as $P \cdot Q^{-1} \pmod{10^9+7}$ for P, Q defined above.

Examples

input

```
2
50 50
50 50
```

output

```
937500007
```

input

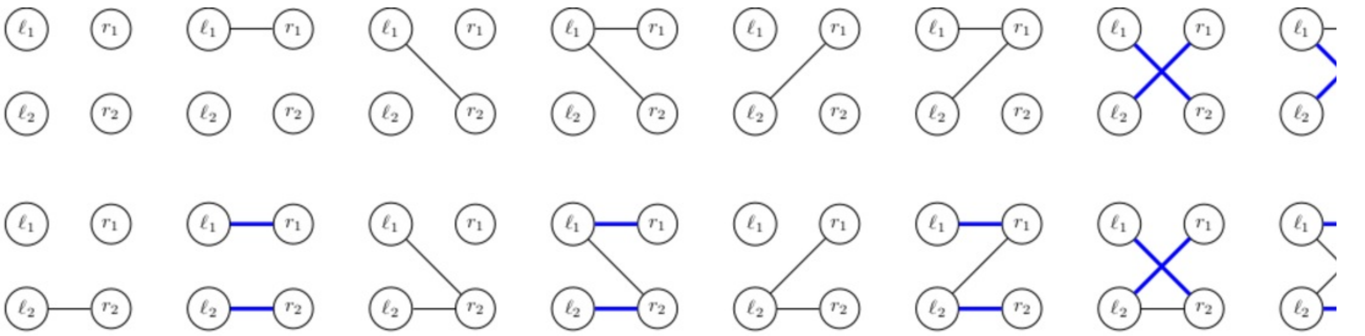
```
3
3 1 4
1 5 9
2 6 5
```

output

```
351284554
```

Note

In the first sample test, each of the 16 graphs below is equally probable. Out of these, 7 have a perfect matching:



Therefore, the probability is equal to $\frac{7}{16}$. As $16 \cdot 562\,500\,004 = 1 \pmod{10^9+7}$, the answer to the testcase is $7 \cdot 562\,500\,004 \mod (10^9+7) = 937\,500\,007$.

G. Mateusz and Escape Room

time limit per test: 7 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mateusz likes to travel! However, on his 42nd visit to Saint Computersburg there is not much left to sightsee. That's why he decided to go to an escape room with his friends!

The team has solved all riddles flawlessly. There is only one riddle remaining — a huge circular table! There are n weighing scales lying on top of the table, distributed along the circle. Each scale is adjacent to exactly two other scales: for each $i \in \{1, 2, \dots, n-1\}$, the i-th and the $(i+1)$-th scales are adjacent to each other, as well as the first and the n-th scale.

The i-th scale initially contains a_i heavy coins. Mateusz can perform moves — each move consists of fetching a single coin from one scale and putting it on any adjacent scale.

It turns out that the riddle will be solved when there is a specific amount of coins on each of the scales. Specifically, each scale has parameters l_i and r_i. If each coin lies on a single scale and for each i, the i-th scale contains at least l_i and at most r_i coins, the riddle will be solved and Mateusz's team will win!

Mateusz is aiming for the best possible time. Therefore, he wants to solved the riddle as quickly as possible. What is the minimum possible number of moves required to fulfill all the conditions?

Input

The first line contains an integer n ($3 \leq n \leq 35\,000$) — the number of weighing scales in the circle.

The following n lines describe the scales. The i-th of these lines describes the i-th scale and consists of three integers a_i, l_i, r_i ($0 \leq a_i \leq 35\,000$, $0 \leq l_i \leq r_i \leq 35\,000$).

It's guaranteed that the riddle is solvable, that is, $\sum_{i=1}^n l_i \leq \sum_{i=1}^n a_i \leq \sum_{i=1}^n r_i$.

Output

Output one integer — the minimum number of operations required to solve the riddle.

Examples

input

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

output

```
4
```

input

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

output

```
1
```

input

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

output

```
0
```