A. Contest

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

Misha and Vasya participated in a Codeforces contest. Unfortunately, each of them solved only one problem, though successfully submitted it at the first attempt. Misha solved the problem that costs a points and Vasya solved the problem that costs b points. Besides, Misha submitted the problem c minutes after the contest started and Vasya submitted the problem d minutes after the contest started. As you know, on Codeforces the cost of a problem reduces as a round continues. That is, if you submit a problem that costs p points t minutes after the contest started, you get $\max\left(\frac{3p}{10}, p - \frac{p}{250} \times t\right)$ points.

Misha and Vasya are having an argument trying to find out who got more points. Help them to find out the truth.

Input

The first line contains four integers a, b, c, d $(250 \le a, b \le 3500, 0 \le c, d \le 180)$.

It is guaranteed that numbers a and b are divisible by 250 (just like on any real Codeforces round).

Output

Output on a single line:

"`Misha`" (without the quotes), if Misha got more points than Vasya.

"`Vasya`" (without the quotes), if Vasya got more points than Misha.

"`Tie`" (without the quotes), if both of them got the same number of points.

Sample test(s)

input
```
500 1000 20 30
```
output
```
Vasya
```

input
```
1000 1000 1 1
```
output
```
Tie
```

input
```
1500 1000 176 177
```
output
```
Misha
```

B. Misha and Changing Handles

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

Misha hacked the Codeforces site. Then he decided to let all the users change their handles. A user can now change his handle any number of times. But each new handle must not be equal to any handle that is already used or that was used at some point.

Misha has a list of handle change requests. After completing the requests he wants to understand the relation between the original and the new handles of the users. Help him to do that.

Input

The first line contains integer q ($1 \le q \le 1000$), the number of handle change requests.

Next q lines contain the descriptions of the requests, one per line.

Each query consists of two non-empty strings old and new, separated by a space. The strings consist of lowercase and uppercase Latin letters and digits. Strings old and new are distinct. The lengths of the strings do not exceed 20.

The requests are given chronologically. In other words, by the moment of a query there is a single person with handle old, and handle new is not used and has not been used by anyone.

Output

In the first line output the integer n — the number of users that changed their handles at least once.

In the next n lines print the mapping between the old and the new handles of the users. Each of them must contain two strings, old and new, separated by a space, meaning that before the user had handle old, and after all the requests are completed, his handle is new. You may output lines in any order.

Each user who changes the handle must occur exactly once in this description.

Sample test(s)

input

```
5
Misha ILoveCodeforces
Vasya Petrov
Petrov VasyaPetrov123
ILoveCodeforces MikeMirzayanov
Petya Ivanov
```

output

```
3
Petya Ivanov
Misha MikeMirzayanov
Vasya VasyaPetrov123
```

C. Misha and Forest

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

Let's define a forest as a non-directed acyclic graph (also without loops and parallel edges). One day Misha played with the forest consisting of n vertices. For each vertex v from 0 to $n - 1$ he wrote down two integers, $degree_v$ and s_v, were the first integer is the number of vertices adjacent to vertex v, and the second integer is the XOR sum of the numbers of vertices adjacent to v (if there were no adjacent vertices, he wrote down 0).

Next day Misha couldn't remember what graph he initially had. Misha has values $degree_v$ and s_v left, though. Help him find the number of edges and the edges of the initial graph. It is guaranteed that there exists a forest that corresponds to the numbers written by Misha.

Input

The first line contains integer n ($1 \le n \le 2^{16}$), the number of vertices in the graph.

The i-th of the next lines contains numbers $degree_i$ and s_i ($0 \le degree_i \le n - 1$, $0 \le s_i < 2^{16}$), separated by a space.

Output

In the first line print number m, the number of edges of the graph.

Next print m lines, each containing two distinct numbers, a and b ($0 \le a \le n - 1$, $0 \le b \le n - 1$), corresponding to edge (a, b).

Edges can be printed in any order; vertices of the edge can also be printed in any order.

Sample test(s)

input

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

output

```
2
1 0
2 0
```

input

```
2
1 1
1 0
```

output

```
1
0 1
```

Note

The XOR sum of numbers is the result of bitwise adding numbers modulo 2. This operation exists in many modern programming languages. For example, in languages C++, Java and Python it is represented as "^", and in Pascal — as "xor".

D. Misha and Permutations Summation

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

Let's define the sum of two permutations p and q of numbers $0, 1, ..., (n - 1)$ as permutation $Perm((Ord(p) + Ord(q)) \bmod n!)$, where $Perm(x)$ is the x-th lexicographically permutation of numbers $0, 1, ..., (n - 1)$ (counting from zero), and $Ord(p)$ is the number of permutation p in the lexicographical order.

For example, $Perm(0) = (0, 1, ..., n - 2, n - 1)$, $Perm(n! - 1) = (n - 1, n - 2, ..., 1, 0)$

Misha has two permutations, p and q. Your task is to find their sum.

Permutation $a = (a_0, a_1, ..., a_{n-1})$ is called to be lexicographically smaller than permutation $b = (b_0, b_1, ..., b_{n-1})$, if for some k following conditions hold: $a_0 = b_0, a_1 = b_1, ..., a_{k-1} = b_{k-1}, a_k < b_k$.

Input

The first line contains an integer n $(1 \le n \le 200\,000)$.

The second line contains n distinct integers from 0 to $n - 1$, separated by a space, forming permutation p.

The third line contains n distinct integers from 0 to $n - 1$, separated by spaces, forming permutation q.

Output

Print n distinct integers from 0 to $n - 1$, forming the sum of the given permutations. Separate the numbers by spaces.

Sample test(s)

input

```
2
0 1
0 1
```

output

```
0 1
```

input

```
2
0 1
1 0
```

output

```
1 0
```

input

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

output

```
1 0 2
```

Note

Permutations of numbers from 0 to 1 in the lexicographical order: $(0, 1), (1, 0)$.

In the first sample $Ord(p) = 0$ and $Ord(q) = 0$, so the answer is $Perm((0 + 0) \bmod 2) = Perm(0) = (0, 1)$.

In the second sample $Ord(p) = 0$ and $Ord(q) = 1$, so the answer is $Perm((0 + 1) \bmod 2) = Perm(1) = (1, 0)$.

Permutations of numbers from 0 to 2 in the lexicographical order: $(0, 1, 2), (0, 2, 1), (1, 0, 2), (1, 2, 0), (2, 0, 1), (2, 1, 0)$.

In the third sample $Ord(p) = 3$ and $Ord(q) = 5$, so the answer is $Perm((3 + 5) \bmod 6) = Perm(2) = (1, 0, 2)$.

E. Misha and Palindrome Degree

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

Misha has an array of n integers indexed by integers from 1 to n. Let's define *palindrome degree* of array a as the number of such index pairs $(l, r)(1 \le l \le r \le n)$, that the elements from the l-th to the r-th one inclusive can be rearranged in such a way that the **whole** array will be a palindrome. In other words, pair (l, r) should meet the condition that after some rearranging of numbers on positions from l to r, inclusive (it is allowed not to rearrange the numbers at all), for any $1 \le i \le n$ following condition holds: $a[i] = a[n - i + 1]$.

Your task is to find the *palindrome degree* of Misha's array.

Input

The first line contains integer n ($1 \le n \le 10^5$).

The second line contains n positive integers $a[i]$ ($1 \le a[i] \le n$), separated by spaces — the elements of Misha's array.

Output

In a single line print the answer to the problem.

Sample test(s)

input
```
3
2 2 2
```
output
```
6
```

input
```
6
3 6 5 3 3 5
```
output
```
0
```

input
```
5
5 5 2 5 2
```
output
```
4
```

Note

In the first sample test any possible pair (l, r) meets the condition.

In the third sample test following pairs $(1, 3), (1, 4), (1, 5), (2, 5)$ meet the condition.