Pair Programming

Input File: *standard input*
Output File: *standard output*

Students are yelling at each other. Hiding under tables. Standing on top of tables. Re-enacting lightsaber duels with pens. Swinging on the ceiling fans... Walking in you are horrified to find your software class wreaking havoc and in a shambles. Shoulders hunched and palm pressed firmly against your face, you attempt to restore some semblance of order by asking some students to program in pairs.

There are N students in your class. Each student writes in a particular programming language, and uses a particular text editor. You've noticed that a pair of students is *cooperative* if they write in the same language or use the same editor, or both.

You would like to write a program that finds the largest number of cooperative pairs you can form, where no student is present in more than one pair. Your program should also output a valid set of pairings (see *Output* and *Scoring* sections below for further details).

Input

The first line of input will contain a single integer N: the number of students in your class. Students are numbered from 1 to N.

N lines follow, the i-th containing two integers l_i e_i: the language and editor that student i uses. Languages and editors are denoted by integers between 1 and N, inclusive.

Output

The first line of output must contain a single integer A: the maximum number of cooperative pairs that you can form.

A lines must follow. Each of these lines must contain two integers identifying two students that should be paired together. No student's number may appear more than once among these A lines. The order of these lines and the order of students within each line does not matter. If there is more than one largest set of pairings, any valid one will be accepted.

See the *Scoring* section below for further details.

Sample Input

```
10
3 6
8 2
4 4
4 7
5 2
5 4
2 1
7 4
9 10
9 10
```

Sample Output

```
4
2 5
8 6
3 4
9 10
```

Explanation

We can pair together

- Student 2 and student 5 (both use editor 2);
- Student 8 and student 6 (both use editor 4);
- Student 3 and student 4 (both write in language 4); and
- Student 9 and student 10 (both write in language 9 and both use editor 10)

for a total of 4 pairs, which is the most we can form in this case. Note that some languages and some editors are not used by any student.

Subtasks & Constraints

For all subtasks, $1 \leq N \leq 1\,000\,000$ and for all $1 \leq i \leq N$, $1 \leq l_i, e_i \leq N$.

- For Subtask 1 (20 points), the language a student writes in has the same number as the editor they use. Specifically, $l_i = e_i$ for all i.
- For Subtask 2 (20 points), $N \leq 50$.
- For Subtask 3 (20 points), $N \leq 1\,000$.
- For Subtask 4 (40 points), no additional constraints apply.

Scoring

For each test case, you will score 100% if the integer A on the first line of output is correct and you provide a correct set of A cooperative pairings. Otherwise, if the integer A is correct, you will score 40% for that test case *even if*:

- You do not supply a set of pairings; or
- You attempt to provide a set of pairings, but it is not a valid set of A cooperative pairs.

Otherwise, you will score 0% for that test case.

Your score for each subtask will be the **minimum** score among all testcases in that subtask. Your score for this problem will be the **maximum** score among all submissions you have made to this problem.