Wabot 2.2

Time Limit: 2.5 seconds

Wabbit has upgraded his Wabot from version 1.9 to 2.2 and now contains more advanced modules. However, more advanced modules also mean more technical issues and thus Wabbit has written up a technical manual to help identify and diagnose these faults.

A certain system in the Wabot has N modules, numbered from 1 to N. Each module is either working or faulty. As a level 1 technician, your job is to analyze a system with exactly one faulty module and determine which module is faulty. To do this, you can perform tests on the modules.

A test is represented by an ordered pair (L, R) where $1 \leq L \leq R \leq N$, and checks all modules m satisfying $L \leq m \leq R$. A test fails if any of the checked modules are faulty, and passes otherwise. You may perform each test any number of times in any order, and you will know the result (pass/fail) of every test performed.

The technical manual specifies T tests, numbered from 1 to T, that can be performed to determine faulty modules. Test i consists of the pair (L_i, R_i). However, the technical manual is still a draft, and the given tests may not be sufficient to uniquely identify the faulty module.

Wabbit has requested that you add a number of additional tests to the manual. To ensure that the manual is not too long, he wants you to add the minimum number of additional tests such that, when used with the existing tests, it is always possible to uniquely identify the faulty module no matter which module is faulty.

If multiple such sets of tests exist, you may output any of them.

Input Format

The first line of input contains two positive integers N and T.

The next T lines of input contain two positive integers each. The ith line contains L_i and R_i respectively.

Output Format

On the first line of output, print A, the minimum number of additional tests necessary to identify the faulty module.

The next A lines should contain two positive integers each. Each line should describe a single test that should be added to the existing list of tests.

Constraints and Subtasks

For all subtasks
$1 \leq N \leq 10^6$ $0 \leq T \leq 10^6$

Subtask	Points	Constraints
1	**5**	$N, T \leq 8$
2	**10**	$T = 0$
3	**11**	$R_i < L_j$ or $R_j < L_i$ for all $i \neq j$
4	**13**	There exist no $i \neq j$ where $L_i \leq L_j \leq R_j \leq R_i$
5	**22**	$N, T \leq 1000$
6	**15**	$N \leq 10^5$, $T \leq 32$
7	**24**	No further restrictions.

Sample I/O

Input 1	Output 1
2 0	1 1 1

Input 2	Output 2
7 8 1 1 1 2 1 3 1 4 5 7 6 7 7 7 7 7	0

Input 3	Output 3
5 2 2 4 3 3	1 1 3

Explanation

For Sample test case 1, no tests are provided. A single test involving only module 1 is sufficient. If module 1 is faulty, this test will fail and we can infer that module 1 is faulty. If module 2 is faulty, this test will pass and we can infer that module 2 is faulty. Note that a test involving only module 2 would also be accepted.

For Sample test case 2, no new tests need to be added as the given tests are sufficient to uniquely determine the faulty module.