

F. Tourist Reform

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

Berland is a tourist country! At least, it can become such — the government of Berland is confident about this.

There are n cities in Berland, some pairs of which are connected by two-ways roads. Each road connects two different cities. In Berland there are no roads which connect the same pair of cities. It is possible to get from any city to any other city using given two-ways roads.

According to the reform each road will become one-way. It will be oriented to one of two directions.

To maximize the tourist attraction of Berland, after the reform for each city i the value r_i will be calculated. It will equal to the number of cities x for which there is an oriented path from the city i to the city x . In other words, r_i will equal the number of cities which can be reached from the city i by roads.

The government is sure that tourist's attention will be focused on the minimum value of r_i .

Help the government of Berland make the reform to maximize the minimum of r_i .

Input

The first line contains two integers n, m ($2 \leq n \leq 400\,000$, $1 \leq m \leq 400\,000$) — the number of cities and the number of roads.

The next m lines describe roads in Berland: the j -th of them contains two integers u_j and v_j ($1 \leq u_j, v_j \leq n$, $u_j \neq v_j$), where u_j and v_j are the numbers of cities which are connected by the j -th road.

The cities are numbered from 1 to n . It is guaranteed that it is possible to get from any city to any other by following two-ways roads. In Berland there are no roads which connect the same pair of cities.

Output

In the first line print single integer — the maximum possible value $\min_{1 \leq i \leq n} \{r_i\}$ after the orientation of roads.

The next m lines must contain the description of roads after the orientation: the j -th of them must contain two integers u_j, v_j , it means that the j -th road will be directed from the city u_j to the city v_j . Print roads in the same order as they are given in the input data.

Example

input
7 9 4 3 2 6 7 1 4 1 7 3 3 5 7 4 6 5 2 5
output
4 4 3 6 2 7 1 1 4 3 7

5 3
7 4
5 6
2 5