CCCHK '15 J5 - No friends in the same group.

Time limit: 0.6s **Memory limit:** 256M

Alice is the teacher of a class of N students. She is going to divide the students into 2 groups. She has the following requirements:

- Alice knows that some pair of students are very good friends and they must not be assigned to the same group.
- Alice also wants to minimize the difference between the sizes of the two groups.

Let's write a program to help Alice to do the grouping, in which the program should:

- Report `-1` if it is not possible to do the grouping.
- Report the group assigned to each student that fulfills the requirements.

Input Specification

- The first line contains two integers N and M, where N denotes the number of students, and M denotes the number of known relationships between students.
- Following are M lines, and each line contains two integers S and T, which means student S and T are very good friends and they must not be assigned to the same group. Each relationship will only be provided once.

Test cases

- In our test cases, we have $2 \le N \le 1000$ and $1 \le M \le 100\,000$.
- In 30% of the test cases, $N \le 16$.
- In another 20% of the test cases, the number of possible ways to group students $\le 10\,000$.

Output Specification

- If there is no way to do the grouping, output `-1`.
- Otherwise, output a string containing N characters representing the group assigned to each of the N students. If the ith student is assigned to group 1, then the ith character in the string is `1`. Otherwise, the ith character in the string is `2`.
- The string should represent the grouping with minimum difference between the sizes of the two groups. (That is, the difference between the number of 1's and the number of 2's in the output should be minimized.) If there is a tie, output the lexicographically smallest solution.

Sample Input 1

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

Output for Sample Input 1

```
1221
```

Sample Input 2

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

Output for Sample Input 2

```
-1
```

Sample Input 3

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

Output for Sample Input 3

```
122211
```

Explanation

In Sample 1, since $(1, 2)$, $(3, 4)$ and $(1, 3)$ cannot be assigned into the same group, to make the difference minimized, we want 1 and 4 be in the same group, and 2 and 3 be in the other. Among the outputs, 1221 is lexicographically smallest. That is, 1 and 4 are in group 1, and 2 and 3 are in group 2.

In Sample 2, there is no way to do the grouping. The reason is that 1 cannot be in the same group of 2 or 3. But 2 and 3 cannot be in the same group, either.

In Sample 3, 1 cannot be with 2 and 3, and 4 cannot be with 5 and 6. The minimized difference is 0, by group $(1, 5, 6)$ and $(2, 3, 4)$ together.