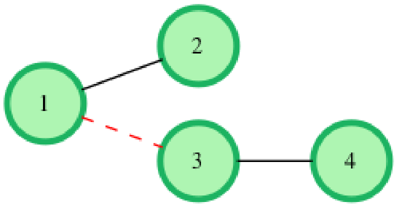
Even Tree

You are given a tree (a simple connected graph with no cycles).

Find the maximum number of edges you can remove from the tree to get a forest such that each connected component of the forest contains an even number of nodes.

As an example, the following tree with 4 nodes can be cut at most 1 time to create an even forest.



Function Description

Complete the *evenForest* function in the editor below. It should return an integer as described.

evenForest has the following parameter(s):

- *t_nodes*: the number of nodes in the tree
- *t_edges*: the number of undirected edges in the tree
- *t_from*: start nodes for each edge
- *t_to*: end nodes for each edge, (Match by index to *t_from*.)

Input Format

The first line of input contains two integers t_nodes and t_edges, the number of nodes and edges. The next t_edges lines contain two integers $t_from[i]$ and $t_to[i]$ which specify nodes connected by an edge of the tree. The root of the tree is node 1.

Constraints

- $2 \leq n \leq 100$
- $n \in \mathbb{Z}^{+}_{\text{even}}$

Note: The tree in the input will be such that it can always be decomposed into components containing an even number of nodes. $\mathbb{Z}^{+}_{\text{even}}$ is the set of positive even integers.

Output Format

Print the number of removed edges.

Sample Input 0

```
10 9
2 1
3 1
```

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

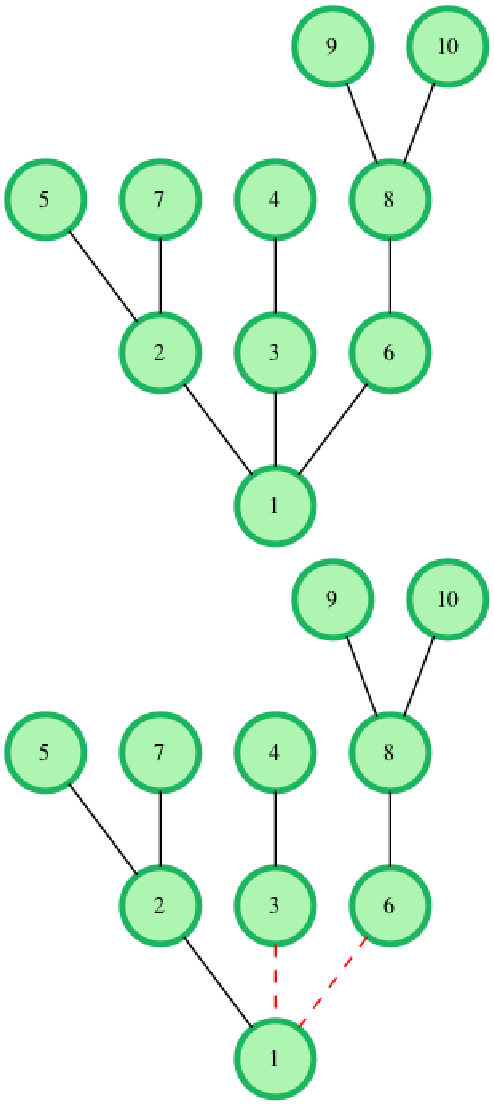
Sample Output 0

```
2
```

Explanation 0

Remove edges $(1, 3)$ and $(1, 6)$ to get the desired result.

. Original tree Decomposed tree



No more edges can be removed.