Description | Hints | Submissions | Discussions | Notes

# Center of Tree

| [illegible] | 1 sec | 256000KB | 100 |
|---|---|---|---|
| Difficulty | Time Limit | Memory | Score |

80/80 XP | 30/30

## Description

You are given a tree consisting of *n* nodes. Find the center of a tree. If there exist multiple centers, print -1.

## Input Format

The first input line contains an integer *n*: the number of nodes. The nodes are numbered 1, 2, ..., *n*.

Then there are *n* - 1 lines describing the edges. Each line contains two integers *a* and *b*: there is an edge between nodes *a* and *b*.

## Output Format

Print the center of a tree. If there exist multiple centers, print -1.

## Constraints

$1 \le n \le 2 \times 10^5$

$1 \le a, b \le n$

### Sample Input 1

Copy

```
3
1 2
1 3
```

### Sample Output 1

Copy

```
1
```

C++14 | 00:00:00 | 12 px

```
56
57      int last_point = 0;
58      for (int i = 1; i <= n; i++)
59      {
60          if (depth[i] > dia_len)
61          {
62              last_point = i;
63              dia_len = depth[i];
64          }
65      }
66      if(dia_len%2==0){
67          //1 center
68          int par=last_point;
69          for(int i=1;i<=dia_len/2;i++){
70              par=parent[par];
71          }
72          cout<<par<<endl;
73      }
74      else{
75          cout<<"-1"<<endl;
76      }
77
78      return 0;
79  }
```

Sample Tests | Manual Tests

Test Case 1 +

Input

```
1
```

Output

Console | Run