Unique Divide And Conquer

HackerRank

Divide-and-Conquer on a tree is a powerful approach to solving tree problems.

Imagine a tree, t, with n vertices. Let's remove some vertex v from tree t, splitting t into zero or more connected components, $t_1, t_2, \ldots, t_k$, with vertices $n_1, n_2, \ldots, n_k$. We can prove that there is a vertex, v, such that the size of each formed components is *at most* $\lfloor \frac{n}{2} \rfloor$.

The Divide-and-Conquer approach can be described as follows:

- Initially, there is a tree, t, with n vertices.
- Find vertex v such that, if v is removed from the tree, the size of each formed component after removing v is *at most* $\lfloor \frac{n}{2} \rfloor$.
- Remove v from tree t.
- Perform this approach recursively for each of the connected components.

We can prove that if we find such a vertex v in linear time (e.g., using *DFS*), the entire approach works in $\mathcal{O}(n \cdot \log n)$. Of course, sometimes there are several such vertices v that we can choose on some step, we can take and remove any of them. However, right now we are interested in trees such that *at each step* there is a unique vertex v that we can choose.

Given n, count the number of tree t's such that the Divide-and-Conquer approach works determinately on them. As this number can be quite large, your answer must be modulo m.

Input Format

A single line of two space-separated positive integers describing the respective values of n (the number of vertices in tree t) and m (the modulo value).

Constraints

- $1 \leq n \leq 3000$
- $n < m \leq 10^9$
- m is a prime number.

Subtasks

- $n \leq 9$ for 40% of the maximum score.
- $n \leq 500$ for 70% of the maximum score.

Output Format

Print a single integer denoting the number of tree t's such that vertex v is unique at each step when applying the Divide-and-Conquer approach, modulo m.

Sample Input 0

```
1 103
```

Sample Output 0

```
1
```

Explanation 0

For $n = 1$, there is only one way to build a tree so we print the value of $1 \bmod 103 = 1$ as our answer.

Sample Input 1

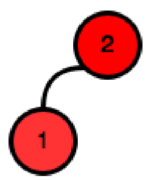
```
2 103
```

Sample Output 1

```
0
```

Explanation 1

For $n = 2$, there is only one way to build a tree:



This tree is *not valid* because we can choose to remove either node 1 or node 2 in the first step. Thus, we print 0 as no valid tree exists.

Sample Input 2

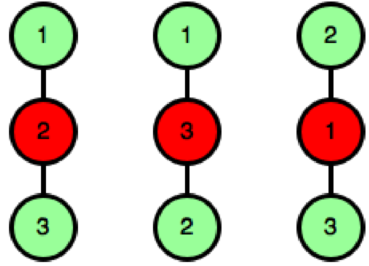
```
3 103
```

Sample Output 2

```
3
```

Explanation 2

For $n = 3$, there are 3 valid trees depicted in the diagram below (the unique vertex removed in the first step is shown in red):



Thus, we print the value of $3 \bmod 103 = 3$ as our answer.

Sample Input 3

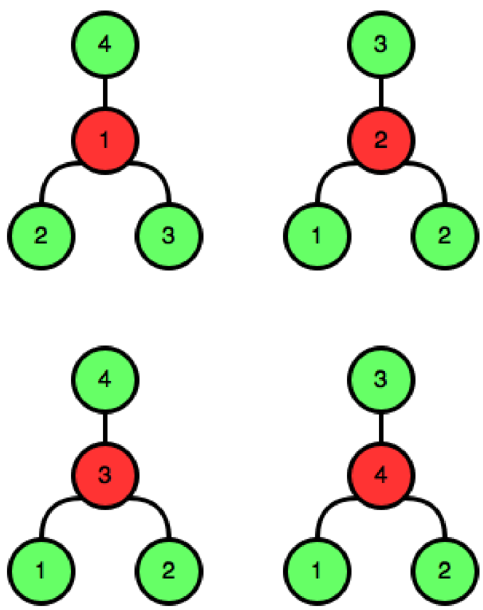
```
4 103
```

Sample Output 3

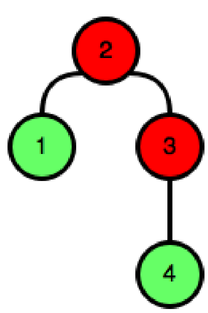
```
4
```

Explanation 3

For $n = 4$, there are 4 valid trees depicted in the diagram below (the unique vertex removed in the first step is shown in red):



The figure below shows an invalid tree with $n = 4$:



This tree is *not valid* because we can choose to remove node 2 or node 3 in the first step. Because we had four valid trees, we print the value of $4 \bmod 103 = 4$ as our answer.