**12.You are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed. All houses at this place are arranged in a circle. That means the first house is the neighbor of the last one. Meanwhile, adjacent houses have security systems connected, and it will automatically contact the police if two adjacent houses were broken into on the same night.**

**Examples:**

**Input: nums = [2, 3, 2]**

**Output: The maximum money you can rob without alerting the**

**police are 3(robbing house 1).**

**(ii) Input: nums = [1, 2, 3, 1]**

**Output: The maximum money you can rob without alerting the**

**police are 4 (robbing house 1 and house 3).**

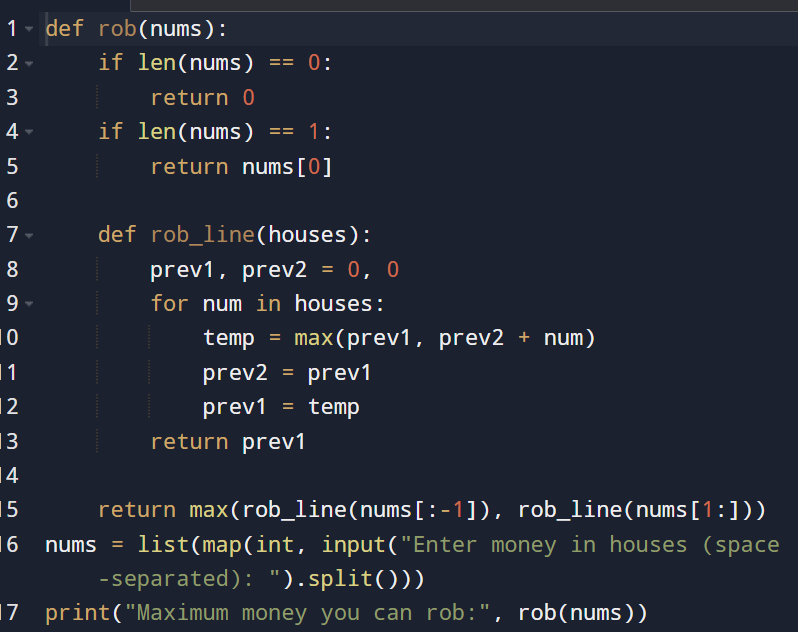
**Aim:**

To find the maximum amount of money a robber can steal from houses arranged in a circle, without robbing two adjacent houses (as it triggers alarms).

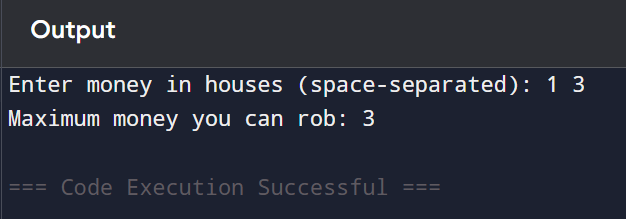
**Algorithm:**

1. Since the houses are in a circle, you cannot rob both the first and last houses.
2. So, split the problem into two linear subproblems:
   * Rob from house 0 to n-2 (excluding the last)
   * Rob from house 1 to n-1 (excluding the first)
3. For each subproblem, apply the standard house robber dynamic programming approach:
   * For each house, decide whether to rob it or skip it, based on previous choices.
   * Use 2 variables to store the maximum robbed amount at each step (space optimization).
4. Return the maximum of the two subproblems.

**Code:**

****

**Input and output:**

****

**Result: given House Robber II – Max Money Robbery Without Alerting Police (Circular Street) is successfully executed and output is verified**

**Performance analysis:**

* **Time Complexity:** O(n)*O*(*n*)
* **Space Complexity:** O(1)*O*(1) (using two variables instead of an array)