📘 Largest Palindrome Product Documentation

1. Problem Statement

You are given an integer n. Your task is to return the largest palindromic number that can be expressed as the product of two n-digit numbers. Because the result can be large, return the value modulo 1337.

Constraints:

- $1 \leq n \leq 8$

2. Intuition

Rather than multiplying every possible pair of n-digit numbers (which would be computationally expensive), we can reverse the process:

- Start from the largest possible palindrome and try to factor it into two n-digit numbers.

3. Key Observations

- The maximum n-digit number is $10^{n - 1.}$
- The minimum n-digit number is $10^{(n-1).}$
- A palindrome is a number that reads the same forwards and backwards.
- If we can generate palindromes efficiently, we can check whether they are the product of two n-digit numbers.

4. Approach

1. Edge Case:
   - If n == 1, the answer is 9, the largest single-digit palindrome.
2. Main Logic:
   - Start from the largest number high = $10^{n - 1.}$

    - Iterate backward for left from high to $10^{(n-1)}$:
        - Generate a palindrome by mirroring left (e.g., 91 → 9009).
        - Check if this palindrome can be factored into two n-digit numbers.
3. Factor Check:
    - Loop from high down to low:
        - If palindrome % i == 0, and the result palindrome // i is also an n-digit number → solution found.
        - Return palindrome modulo 1337.

5. Edge Cases

- n = 1: The largest palindrome formed by multiplying two 1-digit numbers is 9.
- Very large palindromes may not be factorizable within n-digit range, so we must verify both factors.

6. Complexity Analysis

□ Time Complexity:

- Worst-case: $O((10^n)^2)$ for brute force.
- Optimized: $O(N^2)$ where $N = 10^n$, but actual runtime is much less due to early exits and palindrome generation.

□ Space Complexity:

- O(1) → No extra space beyond a few variables.

7. Alternative Approaches

- Brute Force: Check all n-digit combinations, multiply and test for palindrome → Too slow for large n.
- Reverse Check: Start from high product values and check if they are palindromes.
- Precomputed Answers: For interview or contest situations, we can precompute results for n = 1 to 8.

8. Test Cases

Input	Expected Output	Explanation
1	9	1-digit: 3 × 3 = 9
2	987	91 × 99 = 9009 → 9009 % 1337 = 987
3	123 (varies)	Based on largest 3-digit palindrome product

9. Final Thoughts

- This problem is a classic case of optimizing brute-force problems using symmetry (palindromes).
- By reversing the problem—generating palindromes first—we avoid unnecessary computations.
- The solution balances mathematical insight with programming efficiency, making it suitable for competitive programming and interviews.