

# Your Rank is Pure (EXTREME ver)

Note: The problem description is same as [GCJPURE](#), but with higher constraints (to become more challenging), more strict time limit (to reject bad complexity), and more strict source limit (to reject hardcoded precomputation). Good Luck.

---

Pontius: You know, I like this number 127, I don't know why.

Woland: Well, that is an object so pure. You know the prime numbers.

Pontius: Surely I do. Those are the objects possessed by our ancient masters hundreds of years ago. Oh, yes, why then? 127 is indeed a prime number as I was told.

Woland: Not... only... that. 127 is the 31st prime number; then, 31 is itself a prime, it is the 11th; and 11 is the 5th; 5 is the 3rd; 3, you know, is the second; and finally 2 is the 1st.

Pontius: Heh, that is indeed... purely prime.

The game can be played on any subset  $S$  of positive integers. A number in  $S$  is considered pure with respect to  $S$  if, starting from it, you can continue taking its rank in  $S$ , and get a number that is also in  $S$ , until in finite steps you hit the number 1, which is not in  $S$ .

When  $n$  is given, in how many ways you can pick  $S$ , a subset of  $\{2, 3, \dots, n\}$ , so that  $n$  is pure, with respect to  $S$ ? The answer might be a big number, you need to output it modulo  $10^9+7$ .

## Input

The first line of the input gives the number of test cases,  $T$ .  $T$  lines follow. Each contains a single integer  $n$ .

## Output

For each test case, output one line containing "Case #x: y", where  $x$  is the case number (starting from 1) and  $y$  is the answer as described above.

## Constraints

$$T < 10^5$$

$$2 \leq n \leq 10^5$$

Note: These constraints were selected carefully.

## Example

Input:

2

5

6

**Output:**

Case #1: 5

Case #2: 8

## Other Info

Sorry for slow language users, I've made an experiment and the result is if I set constraints that allow slow languages to be accepted with 'good' complexity  $O(f(n))$ , then the 'bad' complexity  $O(f(n)*\log(n))$  could be accepted too using fast language (Because slow language is ~80x slower than fast language). I don't want this to happen. But don't feel so bad :-). I've made [this tutorial problem](#) that allow slow languages to be accepted (except maybe: PIKE).

*Time limit ~4x My Program top speed (25.53s using 1744B of C code).*

*You can see my submission history and time record for this problem: [here](#)*

**See also:** [Another problem added by Tjandra Satria Gunawan](#)