Fibonacci LCM

locked

Problem Submissions Leaderboard Discussions

After Derek (of district 5) discovered how to compute the greatest common divisor (gcd) of Fibonacci numbers, he now tried to answer the next obvious question: how does one compute the *least common multiple* (lcm) of Fibonacci numbers? Unfortunately, Derek found out that this wasn't as easy as the original problem, so he asked you to answer it for him.

The Fibonacci numbers are defined as:

$$F_1 = F_2 = 1$$
$$F_n = F_{n-1} + F_{n-2}$$

Given N integers $a_1, a_2, \ldots, a_N$, find $\text{lcm}(F_{a_1}, F_{a_2}, \ldots, F_{a_N})$, and give your answer modulo $10^9 + 7$.

Input Format

The first line of input contains N.
Each of the next N lines contains a number: the i^{th} line contains a_i.

Constraints

$1 \leq N \leq 100$
$1 \leq a_i \leq 10^9$

Output Format

Print a single integer, which is the least common multiple of the F_{a_i}, modulo $10^9 + 7$.

Sample Input

```
5
1
3
3
6
9
```

Sample Output

```
136
```

Explanation

$\text{lcm}(F_1, F_3, F_3, F_6, F_9) = \text{lcm}(1, 2, 2, 8, 34) = 136$

Submissions: 12
Max Score: 30
Difficulty: Hard

Rate This Challenge:

More

Clojure

```
;
; Complete the 'solve' function below.
;
; The function is expected to return an INTEGER.
; The function accepts INTEGER_ARRAY f as parameter.
;

(defn solve [f]

)

(def fptr (get (System/getenv) "OUTPUT_PATH"))

(def f-count (Integer/parseInt (clojure.string/trim (read-line))))

(def f [])

(doseq [_ (range f-count)]
    (def f (conj f (Integer/parseInt (clojure.string/trim (read-line)))))
)

(def result (solve f))

(spit fptr (str result "\n") :append true)

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code