Bad Paths

Time limit: 1.0s **Memory limit:** 256M

I am currently travelling across a tree with x_i vertices.

A simple path between two distinct vertices is called *good* if every edge in the path belongs to the tree. All other simple paths between two distinct vertices are called *bad* because they contain an edge not found in the tree.

For optimization purposes, I have a plan to create an edge between every pair of vertices, if they are not already directly connected. Since edges are expensive, I will base this decision on the number of **distinct** bad paths. In other words, how many new paths would be created? Please help me calculate this number modulo $10^9 + 7$.

Note: A simple path does not visit the same vertex twice. Two simple paths are considered **distinct** iff there is an edge in one path that is not used in the other path.

Note 2: The exact structure of the tree is irrelevant.

Constraints

For all cases:

$1 \leq N \leq 10^5$

No x_i will appear twice in the same test case.

Subtask	Points	x_i
1	20	$1 \leq x_i \leq 5$
2	30	$1 \leq x_i \leq 100$
3	50	$1 \leq x_i \leq 10^6$

Input Specification

The first line contains the integer N $(1 \leq N \leq 10^5)$.

N lines of input follow. The i^{th} line contains x_i.

Output Specification

For each x_i, output the number of bad paths modulo $10^9 + 7$.

Sample Input

```
2
2
3
```

Sample Output

```
0
3
```