Problem K. NaN in a Heap

Prerequisite: NaN

NaN (Not a Number) is a special floating-point value introduced by the IEEE 754 floating-point standard in 1985. The standard speficies that, when NaN is compared with a floating-point value x (x can be positive, zero, negative, or even NaN itself), the following results should be returned.

Comparison	NaN $\geq$ x	NaN $\leq$ x	NaN $>$ x	NaN $<$ x	NaN $=$ x	NaN $\neq$ x
Result	False	False	False	False	False	True

Prerequisite: Heap

A heap is a data structure which can be represented by a sequence with special properties. The following algorithm demonstrates how to insert n floating-point values $a_1, a_2, \cdots, a_n$ into a min-heap H in order, where H is a sequence and is initially empty.

In the following algorithm, let h_i be the i-th element in sequence H, and let $j/2$ be the maximum integer x satisfying $2x \leq j$.

Algorithm 1 Heapify

```
function HEAPIFY(A)
    Let H be an empty sequence.
    for i ← 1 to n do                ▷ n is the number of elements to be inserted into heap.
        Append a_i to the back of H.
        j := i
        while j > 1 do
            if h_j < h_{j/2} then    ▷ Recall that if h_j or h_{j/2} is NaN, this expression will be false.
                Swap h_j and h_{j/2}.
                j := j/2
            else
                break
            end if
        end while
    end for
    return H
end function
```

Problem

Given an integer n, consider permutations of these n elements: all integers from 1 to $(n-1)$ (both inclusive), as well as a NaN value. We say a permutation P of these n elements is a "heap sequence", if there exists a permutation Q also of these n elements satisfying $P = \texttt{HEAPIFY}(Q)$.

We now randomly pick a permutation of these n elements with equal probability (that is, the probability of a specific permutation to be picked is $\frac{1}{n!}$), calculate the probability that the picked permutation is a heap sequence.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \leq T \leq 10^3$) indicating the number of test cases. For each test case:

The first and only line contains an integer n ($1 \leq n \leq 10^9$).

Output

For each test case output one line containing the answer.

It can be proven that the answer is a rational number $\frac{p}{q}$. To avoid issues related to precisions, please output the integer $(pq^{-1} \bmod M)$ as the answer, where $M = 10^9 + 7$ and q^{-1} is the integer satisfying $qq^{-1} \equiv 1 \pmod M$.

Example

standard input	standard output
5 1 3 7 10 20221218	1 666666672 55555556 596445110 3197361

Note

For the second sample test case, there are 4 heap sequences.

- $\{\texttt{NaN}, 1, 2\} = \texttt{HEAPIFY}(\{\texttt{NaN}, 1, 2\})$.
- $\{\texttt{NaN}, 2, 1\} = \texttt{HEAPIFY}(\{\texttt{NaN}, 2, 1\})$.
- $\{1, \texttt{NaN}, 2\} = \texttt{HEAPIFY}(\{1, \texttt{NaN}, 2\}) = \texttt{HEAPIFY}(\{2, \texttt{NaN}, 1\})$.
- $\{1, 2, \texttt{NaN}\} = \texttt{HEAPIFY}(\{1, 2, \texttt{NaN}\}) = \texttt{HEAPIFY}(\{2, 1, \texttt{NaN}\})$.

So the answer is $\frac{4}{3!} = \frac{2}{3}$ in rational number. As $3 \times 333333336 \equiv 1 \pmod M$, we should output $2 \times 333333336 \bmod M = 666666672$.