

I. Quantum Core Sorting Anomaly

Standard I/O | Time Limit: 2.0s | Memory Limit: 512MB

In the maintenance division of Sector 2026, researchers have uncovered an ancient automated data processor. To stabilize energy crystals, the processor sorts an array $a[1..N]$ using a peculiarly unorthodox routine:

```
for i = 1, 2, ..., N:
  for j = 1, 2, ..., N:
    if a[i] < a[j]:
      swap a[i] and a[j]
```

Here, $\text{swap}(a[i], a[j])$ denotes exchanging the values stored at indices i and j . Despite its strange construction, the routine is mathematically guaranteed to sort any array into non-decreasing order.

The operational inefficiency of the machine on a given permutation is measured by the total number of swaps executed during the sorting procedure. Assuming an initial permutation p of $\{1, 2, \dots, N\}$ is chosen uniformly at random, compute the expected value of the machine's inefficiency.

It can be shown that the expected inefficiency is a rational number A / B . Compute $A \times B^{-1} \bmod 998244353$, where B^{-1} is the modular inverse of B modulo 998244353.

Input

The first line contains a single integer T ($1 \leq T \leq 10^5$) — the number of test cases. Each of the next T lines contains a single integer N ($1 \leq N \leq 10^5$).

Output

For each test case, print a single integer — the expected inefficiency modulo 998244353.

Standard Input	Standard Output
2	166374062
3	504875816
46192	

Explanation for Sample 1 ($N = 3$): There are 6 equally likely permutations of $[1, 2, 3]$:

- $p = [1, 2, 3]$ requires 4 swaps.
- $p = [1, 3, 2]$ requires 3 swaps.
- $p = [2, 1, 3]$ requires 3 swaps.
- $p = [2, 3, 1]$ requires 4 swaps.
- $p = [3, 1, 2]$ requires 2 swaps.
- $p = [3, 2, 1]$ requires 3 swaps.

The expected number of swaps is $(4 + 3 + 3 + 4 + 2 + 3) / 6 = 19 / 6 \equiv 19 \times 6^{-1} \equiv 166374062 \pmod{998244353}$.