Problem A. Algebraic Teamwork

Source file name:	algebraic.c, algebraic.cpp, algebraic.java
Input:	`standard`
Output:	`standard`

The great pioneers of group theory and linear algebra want to cooperate and join their theories. In group theory, permutations – also known as bijective functions – play an important role. For a finite set A, a function $\sigma : A \to A$ is called a permutation of A if and only if there is some function $\rho : A \to A$ with

$$\sigma(\rho(a)) = a \text{ and } \rho(\sigma(a)) = a \text{ for all } a \in A.$$

The other half of the new team – the experts on linear algebra – deal a lot with idempotent functions. They appear as projections when computing shadows in 3D games or as closure operators like the transitive closure, just to name a few examples. A function $p : A \to A$ is called idempotent if and only if

$$p(p(a)) = p(a) \text{ for all } a \in A.$$

To continue with their joined research, they need your help. The team is interested in non-idempotent permutations of a given finite set A. As a first step, they discovered that the result only depends on the set's size. For a concrete size $1 \leq n \leq 10^5$, they want you to compute the number of permutations on a set of cardinality n that are **not** idempotent.

Input

The input starts with the number $t \leq 100$ of test cases. Then t lines follow, each containing the set's size $1 \leq n \leq 10^5$.

Output

Output one line for every test case containing the number modulo $1000000007 = (10^9 + 7)$ of **non-idempotent** permutations on a set of cardinality n.

Example

Input	Output
3 1 2 2171	0 1 6425