Excellent Numbers 2 (`excellent2`)

A positive integer is called *excellent* if its decimal representation contains only the digits 1 and 5, and it is divisible by 3. For example, **15** and **111** are *excellent* numbers ($15 = 5 \cdot 3 + 0$ and $111 = 37 \cdot 3 + 0$), while **151** is not ($151 = 50 \cdot 3 + 1$).

Figure 1: 151515 is the hex triplet[2]of a variation of the *Eerie Black* color and also happens to be an *excellent* number!

Alex observed that there are many excellent numbers formed by N digits, and thus he started counting them. However, this is taking too much time, so he gave this task as a homework for you: help him count how many excellent numbers of N digits exist! Since the answer can be big, print it modulo $10^9 + 7$.

> ☞ Among the attachments of this task you may find a template file `excellent2.*` with a sample incomplete implementation.

Input

The first line of the input file contains a single integer T, the number of test cases. T test cases follow, each preceded by an empty line.

Each test case consists of:

- a line containing 64-bit integer N, representing the number of digits for which we have to find the answer.

[2]The *hex triplet* is a way to represent colors as its Red, Green, and Blue components with two hex digits (one byte) for each component.

Output

The output file must contain T lines corresponding to the test cases, each consisting of integer `ans`, representing the number of excellent numbers with N_i digits modulo $10^9 + 7$ (i.e. the reminder of the division by $10^9 + 7$).

> ✍ The *modulo* operation ($a \bmod m$) can be written in C/C++/Java/Python as `(a % m)`. To avoid the *integer overflow* error, remember to reduce all partial results through the modulus, and not just the final result!
> *Notice that if* $x < 10^9 + 7$*, then* $2x$ *fits into a* C/C++/Java `int`.

Constraints

- $1 \leq T \leq 10$.
- $1 \leq N \leq 10^{18}$.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

– **Subtask 1** (0 points) Examples.

– **Subtask 2** (13 points) $N \leq 20$.

– **Subtask 3** (24 points) $N \leq 2000$.

– **Subtask 4** (34 points) $N \leq 200\,000$.

– **Subtask 5** (29 points) No additional limitations.

Examples

input	output
5 5 3 10 39 952	10 2 342 251936681 897205658

Explanation

In the **first testcase** of the sample case, we have $N = 5$. There are 10 excellent numbers with 5 digits. In increasing order they are: 11115, 11151, 11511, 15111, 15555, 51111, 51555, 55155, 55515 and 55551.

In the **second testcase** we have $N = 3$. There are 2 excellent numbers with 3 digits: 115 and 555.

In the **fourth testcase** there are $183\,251\,937\,962$ excellent numbers with $N = 39$ digits. This number modulo $10^9 + 7$ is $251\,936\,681$.