

Ignore the Garbage

Fred works as an IT consultant in an insurance company. As they always had a large amount of customers waiting and arguing at the front desk, management decided to deploy a ticket machine. Each customer would get a ticket with a number and there will be fancy LCD display over each desk showing the number of the next person. Fred was appointed to get this new enhancement working.

Because Fred is lazy when it comes to manual labor and as an IT consultant he wouldn't lower himself to the level of some hardware technician (except when upgrading his own computer), he asked few technicians to install the displays and prepared himself just to plug in the ticket machine and try it out. Unfortunately (for Fred) the technicians, either inspired by Mr.Bean or because of their carelessness, installed the display upside-down.

Being a software guy, Fred decided that the hardware should not be tampered with after it is installed (except for the case if he would be able to get back the technicians to repair it, but they were already angry at him for his nagging). Then he noted that from time to time the display shows a correct number even when it is upside-down. And hey, the ticket machine is an embedded device and contains a small processor! It would be just a sin for an IT guy not to try to meddle with it and try running an own version of Linux. Now we just need to figure out which readable numbers will the display show.

Task specification

In the beginning the display shows the number 1 on its display. Each second the number shown is increased by 1. We see the display upside-down and thus not everything we see will make sense. Your task is to compute the K-th valid number we will see on the display. The digits the display uses are shown on the images below. An upside-down 1 still count as 1. The number we see may have leading zeroes – e.g. turning the number 600 upside down leads to a valid number.



Input

t - the number of test cases [$t \leq 2200$], than t test cases follows. Each test case consist of one integer K_i [$0 < K_i \leq 10^{200}$].

Output

For each K_i from the input file, output the K_i -th number shown on the display (including the leading zeroes, if there are some).

Example

Input:

8
1
2
3
4
5
6
8
98

Output:

1
2
5
9
8
6
11
002