Problem D: Magic

inputfile: D.IN
outputfile: D.OUT

Description

A well known big computer factory is involved in a very secret project. The project is so secret that we cannot tell you the name of the manufacturer, but we can tell you something about the project. It is called: the Non-Deterministic Magic Holistic New Age Computer (NDMHNAC). It is a major breakthrough in thinking about computing, in that it steps away from every traditional idea in Computer Science and Hardware Design. One of the (many) abilities of NDMHNAC is to execute calculations with very large numbers very fast.

Unfortunately it has a small flaw. Its circuits are based on Magic, so its calculations are disturbed by magic in the same way as traditional circuits are disturbed by electromagnetic fields. As you know the numbers 3 and 7 have strong magic. Therefore these numbers should be avoided, because they will disturb, and eventually destroy the circuits of the machine. The algorithms used internally in the machine never generate offending numbers. The only problem is to ban offending numbers from input.

To that purpose a conventional machine is used as a pre-processor. This machine should transform offending numbers to admissible numbers. (You might wonder how a computer can give a correct answer if you jumble its input. Now this question shows how much you are tied to traditional thinking in computer science. A Non-Deterministic Magic Holistic New Age Computer will give correct answers, even if the input is wrong! (Compare this to traditional computers giving wrong answers, even if the input is correct!)).

Numbers are arbitrarily large, non-negative integers, in decimal representation. A number is offending if it has one of the following properties:
it is divisible by 3, or by 7 (e.g. 6, 14, 21),
a 3 or 7 occurs in its decimal representation, (e.g. 13, 27, 37),
a digit is repeated 3 or 7 times consecutively in its decimal representation. (e.g. 2411145, 10000000).
(NB the number 0 turns out not to be offending).

Offending numbers are to be transformed into non-offending numbers, using (repeatedly) the following rules:
if a number is divisible by 3: divide it by 3
if a number is divisible by 7: divide it by 7
if a 3 occurs in the decimal representation: remove this 3 (remove if possible leading zeros)
if a 7 occurs in the decimal representation: remove this 7 (remove if possible leading zeros)
if a substring of 3 times the same digit occurs in the decimal representation: remove it (remove if possible leading zeros)
if a substring of 7 times the same digit occurs in the decimal representation: remove it (remove if possible leading zeros)
if eventually no digit is left, consider this as the number 0.

A traditional programmer is hired to implement these rules. After reading the specification given above, he complains that it is highly ambiguous. For example, the third rule is about removing a 3. Should all occurrences of 3 be removed, or just any, or only the first one? Moreover, should every rule be applied repeatedly, or should all rules be applied, and then all rules again, and in which order? Different interpretations of the specification may give rise to thousands of different answers.

The designers of the new machine declare that they do not care, as long as a non-offending number is obtained. Seeing that the programmer feels really uncomfortable, they say: 'OK, just give us the largest possible answer'. The programmer got himself an other job.

Problem

Write a program which, given a number, generates the largest non-offending number that can be obtained by applying the rules given above.

Input

The first line of the input-file contains the number of problems *n*. The next *n* lines contain one number each. An input line is never longer than *21* characters ($21 = 3 \times 7$). An input line never starts with *0*.

Output

The outputfile consists of *n* lines with on each line the largest non-offending number that can be obtained by applying the rules given above.

Example

Sample input

```
3
999
273
2331
```

Correct output for the sample input

```
11
2
11
```