

Balanced base-3

[Due to SPOJ restrictions, this problem has been modified with respect to the original version used in the Argentinian Programming Tournament of 2014 in order to have multiple test cases per input file. The original version of this problem (in Spanish) can be found at <http://dc.uba.ar/events/icpc/download/problems/tap2014-problems.pdf>]

Throughout history many different numbering systems have been developed. Some, like Roman numerals, have now been almost completely abandoned due to their lack of convenience. Other even more exotic numbering systems are only ever used in certain applications, such as the factoradic numbers in the context of permutation numbering. In this problem we will consider a numbering system known as *balanced base-3*, which naturally appears in the analysis of diverse mathematical problems related to balance scales.

The balanced base-3 system is similar to the decimal or *base-10* system we are familiar with in that it is *positional*. Positional numbering systems have *digits* whose relative order determines which power of the *base* accompanies them. For example, in base-10 we have

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0.$$

In standard positional systems, the allowed digits are the numbers from **0** to **B-1**, where **B** is the base of the corresponding system. Therefore, the number **123** in base-10 is written in standard base-3 as "**11120**" because

$$1 \times 3^4 + 1 \times 3^3 + 1 \times 3^2 + 2 \times 3^1 + 0 \times 3^0 = 123.$$

Balanced base-3 is different from standard base-3 only in the fact that the allowed digits are **0**, **1** and **-1**, which we will respectively denote as '**0**', '**+**' and '**-**'. Then, the number **123** in base-10 is written as "**+----0**" in balanced base-3, for we have

$$1 \times 3^5 + (-1) \times 3^4 + (-1) \times 3^3 + (-1) \times 3^2 + (-1) \times 3^1 + 0 \times 3^0 = 123.$$

Conversion from numbers in base-10 to balanced base-3 is a mechanical and somewhat tedious process, so we require a program to do it for us. Can you help?

Input

The first line contains an integer number **T**, the number of test cases ($1 \leq T \leq 200$). **T** test cases follow.

Each test case is described by a single line containing a positive integer number **N**, the number in base-10 that we want to write in balanced base-3 ($1 \leq N \leq 1000$).

Output

For each test case, print a single line containing a string composed solely of the characters '**0**', '**+**'

and '-' and not beginning with '0', representing the digits of the number **N** in balanced base-3. Note that the restriction that the string does not start with a '0' ensures that this representation is unique.

Example

Input:

2

123

729

Output:

+----0

+000000