

B. Restoring IPv6

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

An IPv6-address is a 128-bit number. For convenience, this number is recorded in blocks of 16 bits in hexadecimal record, the blocks are separated by colons — 8 blocks in total, each block has four hexadecimal digits. Here is an example of the correct record of a IPv6 address: "0124:5678:90ab:cdef:0124:5678:90ab:cdef". We'll call such format of recording an IPv6-address *full*.

Besides the full record of an IPv6 address there is a *short* record format. The record of an IPv6 address can be shortened by removing one or more leading zeroes at the beginning of each block. However, each block should contain at least one digit in the short format. For example, the leading zeroes can be removed like that: "a56f:00d3:0000:0124:0001:f19a:1000:0000" → "a56f:d3:0:0124:01:f19a:1000:00". There are more ways to shorten zeroes in this IPv6 address.

Some IPv6 addresses contain long sequences of zeroes. Continuous sequences of 16-bit zero blocks can be shortened to " : : ". A sequence can consist of one or several **consecutive blocks**, with all 16 bits equal to 0.

You can see examples of zero block shortenings below:

- "a56f:00d3:0000:0124:0001:0000:0000:0000" → "a56f:00d3:0000:0124:0001::";
- "a56f:0000:0000:0124:0001:0000:1234:0ff0" → "a56f::0124:0001:0000:1234:0ff0";
- "a56f:0000:0000:0000:0001:0000:1234:0ff0" → "a56f:0000::0000:0001:0000:1234:0ff0";
- "a56f:00d3:0000:0124:0001:0000:0000:0000" → "a56f:00d3:0000:0124:0001::0000";
- "0000:0000:0000:0000:0000:0000:0000:0000" → "::".

It is not allowed to shorten zero blocks in the address more than once. This means that the short record can't contain the sequence of characters " : : " more than once. Otherwise, it will sometimes be impossible to determine the number of zero blocks, each represented by a double colon.

The format of the record of the IPv6 address after removing the leading zeroes and shortening the zero blocks is called *short*.

You've got several short records of IPv6 addresses. Restore their full record.

Input

The first line contains a single integer n — the number of records to restore ($1 \leq n \leq 100$).

Each of the following n lines contains a string — the short IPv6 addresses. Each string only consists of string characters "0123456789abcdef:".

It is guaranteed that each short address is obtained by the way that is described in the statement from some full IPv6 address.

Output

For each short IPv6 address from the input print its full record on a separate line. Print the full records for the short IPv6 addresses in the order, in which the short records follow in the input.

Examples

input
6
a56f:d3:0:0124:01:f19a:1000:00
a56f:00d3:0000:0124:0001::

a56f::0124:0001:0000:1234:0ff0
a56f:0000::0000:0001:0000:1234:0ff0
::
0ea::4d:f4:6:0

output

a56f:00d3:0000:0124:0001:f19a:1000:0000
a56f:00d3:0000:0124:0001:0000:0000:0000
a56f:0000:0000:0124:0001:0000:1234:0ff0
a56f:0000:0000:0000:0001:0000:1234:0ff0
0000:0000:0000:0000:0000:0000:0000:0000
00ea:0000:0000:0000:004d:00f4:0006:0000