C. Beautiful IP Addresses

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

The problem uses a simplified TCP/IP address model, please read the statement carefully.

An IP address is a 32-bit integer, represented as a group of four decimal 8-bit integers (without leading zeroes), separated by commas. For example, record `0.255.1.123` shows a correct IP address and records `0.256.1.123` and `0.255.1.01` do not. In the given problem an arbitrary group of four 8-bit integers is a correct IP address.

Our hero Polycarpus still works as a system administrator in some large corporation. He likes beautiful IP addresses. To check if some IP address is beautiful, he should do the following:

1. write out in a line four 8-bit numbers of the IP address, without the commas;
2. check if the resulting string is a palindrome.

Let us remind you that a palindrome is a string that reads the same from right to left and from left to right.

For example, IP addresses `12.102.20.121` and `0.3.14.130` are beautiful (as strings "`1210220121`" and "`0314130`" are palindromes), and IP addresses `1.20.20.1` and `100.4.4.1` are not.

Polycarpus wants to find all beautiful IP addresses that have the given set of digits. Each digit from the set must occur in the IP address at least once. IP address must not contain any other digits. Help him to cope with this difficult task.

Input

The first line contains a single integer n $(1 \le n \le 10)$ — the number of digits in the set. The second line contains the set of integers $a_1, a_2, ..., a_n$ $(0 \le a_i \le 9)$. It is guaranteed that all digits in the set are distinct.

Output

In the first line print a single integer k — the number of beautiful IP addresses that contain the given set of digits. In the following k lines print the IP addresses, one per line in the arbitrary order.

Examples

input
6 0 1 2 9 8 7

output
6 78.190.209.187 79.180.208.197 87.190.209.178 89.170.207.198 97.180.208.179 98.170.207.189

input
1 4

output
16 4.4.4.4 4.4.4.44 4.4.44.4 4.4.44.44

```
4.44.4.4
4.44.4.44
4.44.44.4
4.44.44.44
44.4.4.4
44.4.4.44
44.4.44.4
44.4.44.44
44.44.4.4
44.44.4.44
44.44.44.4
44.44.44.44
```