IP Blocks
Problem ID: ip

Note: The following is a fictitious depiction of Quora's security system.

Quora has been crawled by various attackers and the team occasionally had to deal with such attackers. To secure the product, Quora decides to allow only known IP addresses and has collected n IPv4 address blocks. However, there were too many IP blocks to store and we want to reduce the storage size by minimizing the set of address blocks. For instance, 172.226.69.220/31 and 172.226.69.222/31 can be merged into 172.226.69.220/30.

The IPv4 address blocks will be provided in Classless Inter-Domain Routing (CIDR) notation. CIDR notation specifies an IP address, a slash ('/') character, and a decimal number. The IP address is a 32-bit integer, split up into 4 groups of 8 bits (octets) separated by periods ('.') for readability. The decimal number is the count of consecutive leading 1-bits (from left to right) in the network mask. For example, the IPv4 block 198.51.100.0/22 represents the 1024 IPv4 addresses from 198.51.100.0 to 198.51.103.255.[1]

Given the IPv4 address blocks, write a program to find a set of blocks which has the same coverage as the input data and minimizes the number of blocks.

Input

Your program will receive input from standard input.

The first line contains a positive integer n representing the number of IPv4 address blocks.

In the following n lines, one IPv4 address block will be provided per line.

Output

Your program should write to standard output.

Print a minimal set of address blocks. Print one address block per line, in any order.

If there are multiple possible answers, print any one of them.

Constraints

- $2 \le n \le 10^5$

Scoring

There are 50 test cases. You will get 2 points per correct test case.

Sample Input 1	Sample Output 1
8 172.224.224.32/31 172.224.224.34/31 172.224.224.36/31 146.75.169.110/31 146.75.169.112/31 146.75.169.114/31 146.75.169.116/31 146.75.169.118/31	146.75.169.110/31 146.75.169.112/29 172.224.224.32/30 172.224.224.36/31

---

[1] https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing