Problem I. International Collegiate Routing Contest

Description

You may know that Bluegao University (formly Bluefly University) is famous of networking technology. One day, their headmaster Yuege received a special router, along with a task about routing table.

In this problem, routing table is a (probably) big table with several items, each item represents a subnet. The router has limited function, it can only deal with two next-hops and one main routing table. Packets will be send to next hop A if there exists a subnet containing the destination of the packet in the main routing table. Otherwise they will be send to next hop B.

You may know that, IPv4 uses 32-bit (four-byte) addresses, which limits the address space to 4294967296 (2^{32}) addresses. IPv4 addresses may be written in any notation expressing a 32-bit integer value, for human convenience, they are most often written in the dot-decimal notation, which consists of four octets of the address expressed individually in decimal and separated by periods. But their binary notation is also very useful. For example, IP address 128.2.142.23 can be expressed in dot-binary notation as 10000000.00000010.10001110.00010111. A subnet is a block of adjacent IP addresses with exactly same binary prefix, and usually written as the first IP address in its address space together with the bit length of prefix, like "202.120.224.0/24". If an IP address is in the range of an subnet, we say that this subnet contains the IP address.

Yuege's task is to invert the behaviour of his router, make all packets currently routed to hop A route to hop B, and vice versa. Also he wants to keep the size of the main routing table as small as possible, for performance.

In short, for a given routing table (i.e. a bunch of subnets), we need to get its "complement", i.e. calculate a **minimum** set of subnets which have no intersection with given subnets, and their union must be the whole IPv4 address space.

Remember that Bluegao University is famous of networking tech, as headmaster of Bluegao University, Yuege definitely knows how to solve such problem, but he is too lazy to code, so he turns to you for help.

Input

The first line of the input gives the number of test cases, T. T test cases follow.

For each test case, the first line contains an integer n ($0 \leq n \leq 30000$), the number of subnets. Next n lines, each contains a subnet in the format of $a.b.c.d/l$, a, b, c, d, l are all integers, $0 \leq a, b, c, d < 256$, $0 \leq l \leq 32$.

Note that even if $l = 32$, the "/32" part should not be omitted. And if $l = 0$, the IP address part must be "0.0.0.0".

Output

For each test case, first output one line "Case #x:", where x is the case number (starting from 1). Then on the second line print an integer n indicates the number of subnets in your answer. Next n lines each line contains a subnet in the same format as input. You may output them in any order.

Samples

Sample Input	Sample Output
3 0 1 0.0.0.0/1 1 128.0.0.0/1	Case #1: 1 0.0.0.0/0 Case #2: 1 128.0.0.0/1 Case #3: 1 0.0.0.0/1