Problem B. Binary Code

Input file: `binary.in`
Output file: `binary.out`

Ben has recently learned about binary prefix codes. A binary code is a set of n distinct nonempty code words s_i, each consisting of `0`s and `1`s. A code is called a *prefix code* if for every $i \neq j$ neither s_i is a prefix of s_j nor s_j is a prefix of s_i. A word x is called a *prefix* of a word w if there exists a possibly empty word y, such that $xy = w$. For example, $x =$ `11` is a prefix of $w =$ `110` and $x =$ `0100` is a prefix of $w =$ `0100`.

Ben found a paper with n lines of binary code in it. However, this paper is pretty old and there are some unreadable characters. Fortunately, each word contains at most one unreadable character.

Ben wants to know whether these n lines could represent a binary prefix code. In other words, can he replace every unreadable character with `0` or `1`, so that the code becomes a prefix code?

Input

The first line contains integer n — the number of code words ($1 \leq n \leq 5 \cdot 10^5$).

Next n lines contain nonempty code word records, one per line. Each code word record consists of "`0`", "`1`" and "`?`" characters. Every code word record contains at most one "`?`" character that represents unreadable character.

The total length of words does not exceed $5 \cdot 10^5$.

Output

Output "`NO`" in the first line if the code cannot be a prefix code.

Otherwise, output "`YES`" in the first line. Next n lines shall contain code words in the same order as the corresponding code word records in the input.

If there are several prefix codes, that could have been written on the paper, output any one.

Examples

binary.in	binary.out
4 00? 0?00 ?1 1?0	YES 000 0100 11 100
3 0100 01?0 01?0	NO