Problem H. Huffman Codes

Input file:	`huffman.in`
Output file:	`huffman.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Huffman codes is a famous family of optimal prefix codes. Let us briefly remind information about them.

Consider alphabet Σ of n characters. Binary prefix code for this alphabet is a mapping $c : \Sigma \to \{0,1\}^*$ which assigns a word of 0-s and 1-s to each character of Σ such that for any $u \neq v$ the code word $c(u)$ is not prefix of the code word $c(v)$. Let the length of the code word $c(a)$ be denoted as $|c(a)|$. Given some text t with $p(a)$ occurrences of character a the prefix code is *optimal* for t if the value $\sum\limits_{a \in \Sigma} |c(a)|p(a)$ is minimal possible.

Huffman algorithm produces optimal prefix code for the given text. It proceeds as follows. If $n = 2$ then clearly the optimal prefix code is 0 for one character and 1 for another. Let $n > 2$. Consider two characters x and y with smallest values of $p(x)$ and $p(y)$, respectively. Unite these characters to a new character z with $p(z) = p(x) + p(y)$. Build optimal prefix code for the resulting $n-1$-character alphabet. Now append 0 to code word for z to get code word for x and append 1 to code word for z to get code word for y.

For example, consider text "abacaba". We have $p(a) = 4$, $p(b) = 2$ and $p(c) = 1$. Unite b and c to one character d to get $p(a) = 4$ and $p(d) = 3$. Now we have $c(a) = 0$ and $c(d) = 1$. Getting back to b and c we have $c(b) = 10$ and $c(c) = 11$. Clearly, Huffman code is not unique, because we are free to swap x and y at any step of the algorithm above to get another Huffman code. In the given example another possible Huffman code is $c(a) = 0$, $c(b) = 11$, $c(c) = 10$.

You are given integers u_i and v_i for i from 1 to n. Find out whether there exists some text t and some Huffman code for it such that the code word for the i-th character contains exactly u_i zeroes and v_i ones. If there exists such text and code you have to print the corresponding Huffman code.

Input

The input file contains multiple test cases.

The first line of each test case contains n ($2 \le n \le 100$). The following n lines contain two integers each: u_i and v_i ($0 \le u_i, v_i \le 100$, $u_i + v_i > 0$).

The last test case is followed by a line containing zero, it must not be processed.

Output

For each test case first output a line containing "`Yes`" if there exists corresponding text t or "`No`" if there exists none. If the answer is positive, the following n lines must contain code words, one on a line. Code words must consist of 0-s and 1-s, the i-th of the printed code words must contain u_i zeroes and v_i ones, the printed code must be Huffman code for some text over n-character alphabet.

If there are several solutions, print any one.

Examples

huffman.in	huffman.out
3 1 0 0 2 1 1 7 1 1 2 1 3 0 1 1 1 2 1 3 0 4 2 1 1 1 1 0	Yes 0 11 10 Yes 10 001 000 01 110 1110 1111 No