

tourist

Input file: `stdin`
Output file: `stdout`
Time limit: 3 seconds
Memory limit: 256 megabytes

The road system of Lemuria is quite complicated. Some of the cities are connected by some kind of one way or two way roads. However, lemurs keep their roads in the proper order and their road maps are always up to date.

G. the tourist just arrived by Penguin Air flight to Lemurian city S . His return flight departs from city F . G. has a list of the most popular attractions of Lemuria and he is eager to visit all of them. Also he has a latest version of the road map, so for every city P he has a list of cities that can be reached by a direct road from P . Your task is to help the tourist to plan the route across Lemuria that starts in S , goes through all cities with attractions (in arbitrary order) and ends in F .

Input

The first line of the input file contains four integers: total number of Lemurian cities N ($1 \leq N \leq 2000$), number of cities with attractions K ($0 \leq K \leq N$), and ordinal numbers of cities S and F (all cities are enumerated from 1). Second line contains K distinct numbers — cities with attractions.

Last N lines contain the road map. Each of these lines contains N characters: if q^{th} character of the $(p+2)^{nd}$ line of input is 1 then there is a direct way from city p to city q , if it is 0 then there is no direct way. There is no road from a city to itself.

Output

If there is no route that satisfies the conditions, output should contain one line with the word “NO”. Otherwise the first line of output should contain the word “YES”, the second line should contain the number of cities in the route M ($M \leq N^2$), and in the following line(s) M numbers should follow — the numbers of cities in the route. The first city in the route should be S , the last city should be F , and for any two adjacent cities in the route, there should be a direct road from the first city to the second one.

Examples

stdin	stdout
4 4 1 2 1 2 3 4 0100 0011 1000 1000	YES 8 1 2 4 1 2 3 1 2
4 2 1 4 2 3 0110 0001 0001 0000	NO
4 1 2 3 4 0010 1001 1000 0100	YES 7 2 4 2 1 3 1 3