

B. New Year Permutation

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

User ainta has a permutation p_1, p_2, \dots, p_n . As the New Year is coming, he wants to make his permutation as pretty as possible.

Permutation a_1, a_2, \dots, a_n is *prettier* than permutation b_1, b_2, \dots, b_n , if and only if there exists an integer k ($1 \leq k \leq n$) where $a_1 = b_1, a_2 = b_2, \dots, a_{k-1} = b_{k-1}$ and $a_k < b_k$ all holds.

As known, permutation p is so sensitive that it could be only modified by swapping two distinct elements. But swapping two elements is harder than you think. Given an $n \times n$ binary matrix A , user ainta can swap the values of p_i and p_j ($1 \leq i, j \leq n, i \neq j$) if and only if $A_{i,j} = 1$.

Given the permutation p and the matrix A , user ainta wants to know the prettiest permutation that he can obtain.

Input

The first line contains an integer n ($1 \leq n \leq 300$) — the size of the permutation p .

The second line contains n space-separated integers p_1, p_2, \dots, p_n — the permutation p that user ainta has. Each integer between 1 and n occurs exactly once in the given permutation.

Next n lines describe the matrix A . The i -th line contains n characters '0' or '1' and describes the i -th row of A . The j -th character of the i -th line $A_{i,j}$ is the element on the intersection of the i -th row and the j -th column of A . It is guaranteed that, for all integers i, j where $1 \leq i < j \leq n$, $A_{i,j} = A_{j,i}$ holds. Also, for all integers i where $1 \leq i \leq n$, $A_{i,i} = 0$ holds.

Output

In the first and only line, print n space-separated integers, describing the prettiest permutation that can be obtained.

Examples

input
<pre>7 5 2 4 3 6 7 1 0001001 0000000 0000010 1000001 0000000 0010000 1001000</pre>
output
<pre>1 2 4 3 6 7 5</pre>
input
<pre>5 4 2 1 5 3 00100 00011 10010 01101 01010</pre>
output
<pre>4 2 1 5 3</pre>

Good Bye 2014

Finished

→ **Virtual participation**

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.




Start virtual contest

→ **Problem tags**

dfs and similar dsu graphs greedy
math sortings

No tag edit access

→ **Contest materials**

- Announcement 
- Tutorial #1 
- Tutorial #2 

Note

In the first sample, the swap needed to obtain the prettiest permutation is: (p_1, p_7) .

In the second sample, the swaps needed to obtain the prettiest permutation is $(p_1, p_3), (p_4, p_5), (p_3, p_4)$.

A **permutation** p is a sequence of integers p_1, p_2, \dots, p_n , consisting of n distinct positive integers, each of them doesn't exceed n . The i -th element of the permutation p is denoted as p_i . The size of the permutation p is denoted as n .