Rebuilding Roads

Problem Statement

In a country called “Dual Pig” (DP), there are numbers of cities. And there are bidirectional roads connecting all of the cities, so people in DP can travel to any city they want. But as the time goes by, some of the roads are destroyed due to improper maintenance leaded by the old king, making some of cities disconnected with other cities. The latest yet greatest king, Bertgod, wants to re-organize the roads on this country. He wants to build some new roads and remove some of the old roads in the country, so there will be **exactly one path** between every pair of distinct cities after the road reconstruction. (Because road maintenance costs a lot, he wants to reduce the number of roads in the country as many as possible, yet the people can still travel to any cities.)
Of course, building new roads and destroying old ones costs a lot of money, Bertgod wants to minimize the total cost spent on the reconstruction.

Input

There are multiple test cases. For each test case, there is an integer **N** (0<N<52) representing the number of cities in the country. Then, there is an N*N matrix **C WITHOUT SPACE** in between, representing the current connection status. C[i][j] will either 1 or 0, C[i][j] = 1 means there is an old road from city i to city j, while 0 means the old road is destroyed. You can assume “C[i][j] = C[j][i] ”, and C[i][i] = 0.
Then, there is another N*N matrix **B** WITH SPACE between elements, representing the building new road cost. B[i][j] means the building cost between city i and city j, You can assume “B[i][j] = B[j][i]”, and B[i][i] = 0.
Finally, there is another N*N matrix **D** WITH SPACE between elements, representing the removal cost. D[i][j] means the road removal cost between city i and city j if there is an old road connecting to them, You can assume “D[i][j] = D[j][i]”, and D[i][i] = 0.

Input ends when N = 0.

Output

For each test case, output the minimum cost spent on the reconstruction.

Examples

Input	**Output**

3
000
000
000
0 1 3
1 0 2
3 2 0
0 1 3
1 0 2
	3
	1
	7
	0
	0
	65

3 2 0
3
011
101
110
0 1 3
1 0 2
3 2 0
0 1 3
1 0 2
3 2 0
6
011000
101000
110000
000011
000101
000110
0 1 3 5 5 5
1 0 2 5 5 5
3 2 0 5 5 5
5 5 5 0 1 3
5 5 5 1 0 2
5 5 5 3 2 0
0 1 3 5 5 5
1 0 2 5 5 5
3 2 0 5 5 5
5 5 5 0 1 3
5 5 5 1 0 2
5 5 5 3 2 0
1
0
0
0
4
0001
0001
0001
1110
0 31 14 35
31 0 28 2
14 28 0 15
35 2 15 0
0 22 5 7	

```
22 0 49 20
5 49 0 21
7 20 21 0
10
0000000000
0000000011
0001010000
0010010000
0000001000
0011000000
0000100000
0000000011
0100000101
0100000110
0 33 15 4 42 36 18 5 12 12
33 0 31 10 15 45 44 3 26 29
15 31 0 50 6 16 36 26 42 13
4 10 50 0 30 11 41 17 41 38
42 15 6 30 0 18 31 13 48 40
36 45 16 11 18 0 39 2 0 10
18 44 36 41 31 39 0 29 9 18
5 3 26 17 13 2 29 0 25 51
12 26 42 41 48 0 9 25 0 39
12 29 13 38 40 10 18 51 39 0
0 32 19 42 22 22 49 4 24 7
32 0 23 15 32 35 51 8 17 0
19 23 0 37 30 19 38 22 47 19
42 15 37 0 16 36 48 49 17 14
22 32 30 16 0 42 32 27 9 9
22 35 19 36 42 0 34 19 51 37
49 51 38 48 32 34 0 46 7 27
4 8 22 49 27 19 46 0 48 36
24 17 47 17 9 51 7 48 0 39
7 0 19 14 9 37 27 36 39 0
0
```