Problem E. Printing Stickers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	30 seconds
Memory limit:	1024 megabytes

You work in the International Collaborative Printing Company (ICPC) which prints documents, booklets, and stickers for all kinds of campaigns. Recently, your company acquired a new sticker printing machine. This printing machine prints pre-configured triangle shades onto a huge sticker sheet. The sheet can be described as a grid of $M \times N$ cells, where M is the number of rows and N is the number of columns. Each cell is divided into two triangles in some specified way, and the machine fills some of the triangles with inks. See Figure 8 as an example.

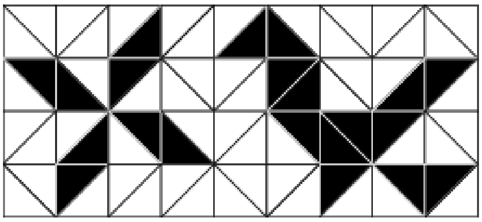

Figure 8: A printed sticker sheet before cutting into pieces of individual stickers.

Since each sticker is relatively small, the printer prints multiple stickers on a sheet, and then cuts them off from the sheet. The cut can be made along the boundary of the triangles. However, to ensure the high quality of a sticker, it is required that all cuts should not be cut along the edge of any filled triangle. Notice that after cutting the sticker sheet into lots of smaller pieces, the pieces without any filled triangles are thrown away. Each remaining sticker has at least one filled triangle. Moreover, on each sticker, all the filled triangles are connected. That is, for any two filled triangles there is a sequence of filled triangles on the same sticker such that every pair of neighboring triangles share at least one common point. Figure 9 demonstrates valid and invalid ways to cut the sticker sheet.

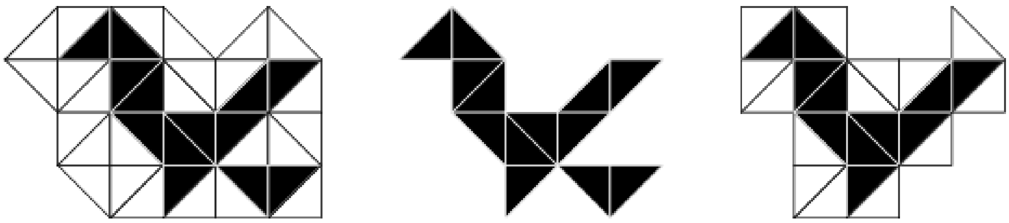

Figure 9: The leftmost sticker is valid, but the other two stickers are invalid.

After cutting off all the stickers from the sheet, an additional polishing step is applied to the boundary of each sticker. Notice that it is possible for a sticker to have holes (see Sample Input 2). In this case, the boundary of the holes also needs to be polished. Cutting the sheet is free, but the polishing step is associated with some costs. Specifically, the boundary of each sticker can be described by a collection of segments. Each segment refers to a horizontal cut, a vertical cut, or a diagonal cut. Each horizontal cut is associated with a polishing cost of H and each vertical cut is associated with a polishing cost of V. The diagonal cuts have slightly different polishing costs $\{D_{ij}\}$, depending on the location of the cut. In particular, the diagonal cut at the ith row and the jth column has a polishing cost D_{ij}. The cost of polishing a sticker is then defined by the sum of all polishing costs associated with the segments. See Figure 10 and Figure 11.

Please write a program that helps the company compute for each sticker, the minimum cost of cutting and then polishing. One can prove that there exists an optimal way to cut off all the stickers from the sticker sheet at once, such that every sticker achieves the minimum polishing cost simultaneously.

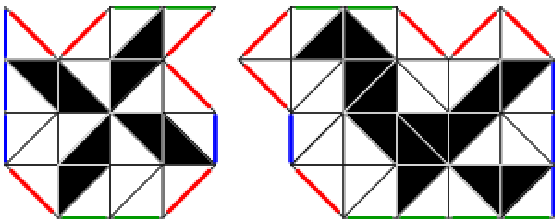

Figure 10: An illustration of computing the cost for polishing stickers. Suppose $H = V = 10$, and each diagonal cut has cost $D_{ij} = 1$. The first sticker on the left is comprised of 4 horizontal segments, 4 vertical segments, and 6 diagonal segments, so the total cost is 86. The second sticker on the right has a cost of 106.

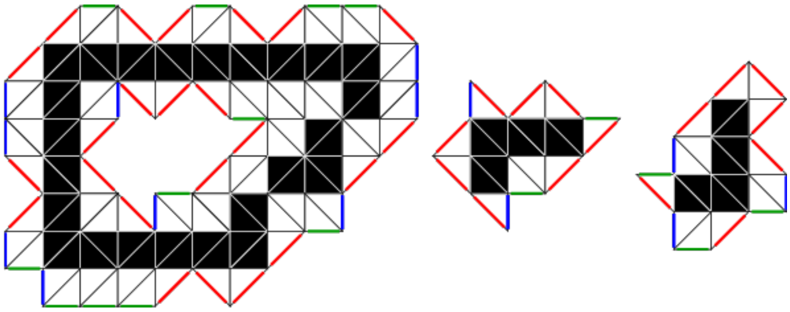

Figure 11: An illustration to Sample Input 2. Notice that the boundary of holes requires polishing. The polishing costs for each sticker are 723, 196, and 214 respectively.

Input

The first line contains an integer T, which represents the number of test cases. For each test case, the first line contains four integers M, N, H, and V, which represent the number of rows, the number of columns, the polishing cost per horizontal segment, and the polishing cost per vertical segment respectively. Then each of the following M lines contains a string of length N, where each character in the string is either '/' or '\'. The jth character of the ith string denotes the direction of the diagonal. Then another M lines follow. In each of the M lines, there is a string of length $2N$. For each cell location (i, j) in the grid, the $(2j + 1)$th character of the ith string denotes whether or not the left-hand-side triangle of the diagonal is filled or not, whereas the $(2j + 2)$th character of the ith string denotes whether or not the right-hand-side triangle of the diagonal is filled. If a triangle is filled with ink then the corresponding character is '#', otherwise, the corresponding character is '.'. Finally, there are M lines. In the ith line there are N integers $D_{i1}, D_{i2}, \ldots, D_{iN}$, denoting the cost of polishing each diagonal.

Constraints

- $1 \le T \le 50$.
- $M \ge 2$; $N \ge 2$; $4 \le M \times N \le 10{,}000$.
- $1 \le H, V, D_{ij} \le 1{,}000$ for all $1 \le i \le M$ and $1 \le j \le N$.
- For each test case, at least 1 and at most 1,000 stickers will be produced from the machine.
- There is no filled triangle whose edge is the boundary of the entire sticker sheet.

Output

Each test case outputs two lines. In the first line, there is an integer k, denoting the total number of stickers. In the second line, there are k integers $c_1, c_2, \ldots, c_k$, denoting the minimum cost for polishing all stickers, sorted in the non-decreasing order.

Examples

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>1
4 9 10 10
\////\\/\
\\/\//\//
//\\/\\/\
\///\//\/
.....#...##.......
.##.#.....##...##.
...#.##....####...
..#.........#..##.
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1</td><td>2
86 106</td></tr>
<tr><td>1
8 12 20 19
\//\//\/\\\/
//\/\\\/////
///\/\\\\//\
\//\\\/\/\/\
\\\\\///\///
///\\\\\\\\\
//\/////\\\\
/////\//////
....
..##################....
..##..............##....
..##..######....##......
..##..##......####..##..
..##........##......##..
..############....####..
....
11 12 13 15 14 12 17 16 14 13 11 10
12 13 15 14 13 17 18 17 15 14 12 16
16 17 18 17 15 14 13 11 16 17 18 19
19 11 12 13 15 14 16 16 17 18 14 13
13 20 16 15 14 14 13 11 10 12 12 13
16 17 17 14 15 16 19 12 14 11 14 16
18 17 14 14 16 19 18 14 15 13 12 14
15 16 17 15 11 18 19 16 16 14 14 20</td><td>3
196 214 723</td></tr>
</table>