Problem B. Shortest Path

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	1024 mebibytes

You are given a grid consisting of $n \times m$ square cells, where each cell is painted in one of two colors: black or white.

You can choose any **black** cell and place a token there. Then, you have to play a game. During every turn of the game, the following happens:

- if the current cell where the token is located has no neighboring (sharing a border) cells of **black** color, the game ends;
- otherwise, you have move the token to one of the neighboring black cells. The cell where the token was located is then repainted to white.

If there are multiple neighboring black cells when you move the token, you can choose which cell to move to. What is the **minimum** number of cells you can visit before the game ends if you act optimally?

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases.

The first line of each test case contains two integers n and m ($2 \le n, m \le 100$) — the number of rows and columns in the grid.

The following n lines describe the grid itself. The i-th line contains a string of m characters; each character is either "`.`" (white cell) or "`X`" (black cell).

Additional constraints on the input:

- there is at least one black cell in each test case;
- the total number of cells across all test cases does not exceed 10^4.

Output

For each test case, print the answer as follows:

- first, print one integer k ($1 \le k \le n \cdot m$) — the minimum number of cells you can visit;
- then, print k pairs of integers x_i, y_i ($1 \le x_i \le n$; $1 \le y_i \le m$) — the cells you visit in the order you visit them.

If there are multiple optimal answers, you may print any of them.

Example

standard input	standard output
4 2 2 X. .X 3 3 .XX XX. XX. 2 3 XXX XXX 4 6 ...XXX XXXX.X X.XXXX XXX...	1 2 2 2 1 2 1 3 4 2 1 2 2 1 2 1 1 8 2 4 3 4 3 5 3 6 2 6 1 6 1 5 1 4