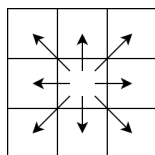


Problem D

Treasure Lair

During your recent exploration, you come across a treasure lair that can be represented as a grid with N rows (numbered from 1 to N) and M columns (numbered from 1 to M). The cell at row r and column c is denoted as (r, c) . Cell (r, c) contains a treasure if $A_{r,c} = 1$, and it is empty if $A_{r,c} = 0$.

There are Q independent scenarios. For each scenario, you start in cell (R, C) and you want to take exactly K treasures. In one second, you can move to any orthogonally or diagonally adjacent cell to the cell you are currently in, as shown in the following illustration. Since the treasures are heavy, you can only carry one treasure at a time, meaning you must bring the treasure back to cell (R, C) before going for the next one or completing the scenario. The action of taking or putting down a treasure takes zero seconds.



For each scenario, determine the minimum required time (in seconds) to take K treasures back to cell (R, C) if you start in (R, C) . As all scenarios are independent of each other, the treasures are back to their original positions at the beginning of a scenario.

Input

The first line consists of two integers $N M$ ($1 \leq N, M \leq 1000$).

Each of the next N lines consists of a binary string A_i of length M . The j^{th} character of string A_i describes cell (i, j) : it is 1 if cell (i, j) contains a treasure, and 0 if cell (i, j) is empty. The number of treasures in the lair is at least one.

The next line consists of an integer Q ($1 \leq Q \leq 10\,000$).

Each of the next Q lines consists of three integers $R C K$ ($1 \leq R \leq N; 1 \leq C \leq M; 1 \leq K$) describing each scenario. The value of K does not exceed the number of treasures in the lair.

Output

For each scenario, output an integer in a single line representing the minimum required time (in seconds) to take K treasures back to cell (R, C) if you start in (R, C) .

Sample Input #1

```
5 5
11010
01001
10001
00111
11010
6
1 1 1
1 1 3
1 1 13
2 3 6
4 1 8
1 5 3
```

Sample Output #1

```
0
4
74
18
32
8
```

Explanation for the sample input/output #1

For scenario 1, you already have 1 treasure at cell (1, 1).

For scenario 2, you can take treasures at cell (1, 1), (1, 2), and (2, 2).

For scenario 3, you need to take all the treasures.

Sample Input #2

```
1 10
0000011111
3
1 1 1
1 1 5
1 10 5
```

Sample Output #2

```
10
70
20
```