Coffee Empire

Coffee Empire

Patras is a city in Greece especially known for one thing, coffee shops. Wherever you go, you will find a coffee shop to buy a coffee "on the go", or to sit and drink a hot or cold one. But let's meet our investor, Kathrin.
Kathrin wants to invest her money and build a new coffee shop. However, to cope with the high competitiveness, she is trying to find the optimal place to build it. Towards this direction, she collected some really detailed statistics. These statistics include the number of coffees that could be bought per corner (for all corners of town) on any given day of week. This number is calculated based on the number of people passing by each corner. The statistics also suggest that all shops sell coffees for the same price, meaning **one euro per coffee** , and that each shop, in order for it to be able to cover the expenses for warming up the appliances and for the employee that serves the customers, it should gain at least 20 euros for any day that is open (otherwise it should remain closed).
Based on these, Kathrin found a simple way to calculate the income for any given corner, which she plans to use in order to find the best spot to open her coffee shop. Let's have the following example with named corners:
A B C
D E F
G H I
First we define the notion of **adjacent corners** : the adjacent corners of a specific corner are the ones that are positioned in cross—like shape with regards to the corner at question. For example, the adjacent corners of corner E would be the ones labeled with B, D, F and H whereas the adjacent corners for corner C would be the ones labeled with B and F.
Based on the definition above and according to Kathrins plan, the expected future income ***for our coffee shop*** and for a given day and a given corner should be calculated according to the following rules:

- The daily income for that corner would equal to the respective value according to the detailed statistics **plus** any additional income that might derive from its adjacent corners (see rule below) .
- If we were to build our coffee shop at a particular corner, then we should also take into account additional expected daily income deriving from coffees that may be sold due to traffic from the ***adjacent corners*** of our corner. In particular, for every adjacent corner of the corner at question that **a)** has a daily income (**as reported by the statistics**) greater or equal to 5 Euros **and b)** has **exactly 3 coffee shops** already located in it, we can safely assume **an extra daily income of 1 Euro** due to that adjacent corner. If either or none of the conditions are not met for an adjacent corner, then we should not expect additional income from it.
- The income corresponding our shop for a given day and corner, will equal the daily income for that corner (**as reported by the statistics plus any additional income based on its adjacent corners**) divided by the overall number of coffee shops located at this corner (note that this value is not always guaranteed to be an integer). For example, if there already exist 2 coffee shops at corner E and the daily income for this corner according to the statistics and the adjacent corners of E equals 27 euros, then we should expect that the daily income for our shop will be 9 euros, as the overall amount of 27 euros will be equally

split to all three coffee shops of this corner (also considering our own shop in case we decide to build it at that specific corner).
- In order for our coffee shop to be open during a given day of the week, the total income corresponding to our coffee shop for that day should be **greater or equal** to its daily expenses (i.e. **20 Euros**). Otherwise, there is no point opening our coffee shop during that day and thus both income and expenses for that day would equal zero.
- According to the town hall rules, it is **forbidden to install more than 3 coffee shops per corner** .

Task

Your task is to develop a program that can find the position that is best for us (i.e. the position that makes us the most profit **on a weekly basis**) to set up our new coffee shop. Can you help us?

Input

Your program will receive the following input from the Standard Input Stream:
In the first line, the width (W<=100) and height (H<=100) of our town, which we can assume that is a perfect grid.
Then H lines will follow each one describing the number of coffee shops already located at each corner. Each line will be composed of a character belonging to the [-, L, M, H] alphabet followed by an asterisk (*), with the exception of the character corresponding to the last corner of each line, which will be followed by a newline character. A dash "-" means "no coffee shops", "L" means only one coffee shop, "M" means two and "H" means three. For example, the first line of the provided sample case indicates that there are no coffee shops located at the first corner (X:1, Y:1), three coffee shops on the second corner (X:2, Y:1) and no coffee shops on the third corner (X:3, Y:1).
Then seven [W, H] blocks of lines will follow, each one corresponding to the statistics of each corner for each day of the week. In the first line of each block, the name of the day will be given followed by the number of coffees that will be sold per corner. Again, the number of coffees sold per corner will be separated by an asterisk (*) with the exception of the number corresponding to the last corner of each line, which will be followed by a newline character.

Output

Your program should output to the Standard Output Stream the coordinates where we should place our new coffee shop. The coordinates are considered to start from 1 and should be reported in the [X, Y] order, where X corresponds to a column and Y to a line of the town grid. These two values should be separated by a single space character. If there are more than one corners with as equally as optimal income, then the first should be printed. In case, there is no such a place, then the output “-1 -1” (without the quotes) should be printed on the screen.
Note: There is NOT a newline character at the end of the output stream

Sample Input 1

```
3 3
```

```
-*H*-
M*M*-
H*-*L
Monday
30*1*1
20*13*1
0*0*0
Tuesday
0*2*2
0*1*1
1*1*1
Wednesday
0*0*2
1*1*3
0*0*1
Thursday
0*1*1
0*1*0
0*1*1
Friday
0*0*0
0*2*3
0*1*1
Saturday
0*2*0
1*3*0
0*1*2
Sunday
0*2*3
0*1*0
1*0*0
```

Sample Output 1

```
1 1
```