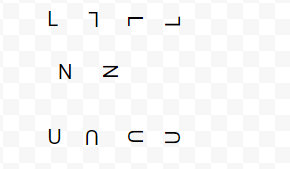
**Task 1.** Your task is to interpret a letter represented in a 3x3 matrix of cells using zeros and ones. Below are three sample 3x3 letter representations for the problem:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L | | |  | N | | |  | U | | |
| 1 | 0 | 0 |  | 1 | 0 | 1 |  | 1 | 0 | 1 |
| 1 | 0 | 0 |  | 1 | 1 | 1 |  | 1 | 0 | 1 |
| 1 | 1 | 1 |  | 1 | 0 | 1 |  | 1 | 1 | 1 |

Each of these representations maintains the intended letter when rotated in any direction. For example, the L matrix can be read as stated or rotated, and the shape still represents the same letter.

The following displays all possible representations of the letters L, N and U.



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| L | 1 | 0 | 0 |  | 1 | 1 | 1 |  | 1 | 1 | 1 |  | 0 | 0 | 1 |
| 1 | 0 | 0 |  | 0 | 0 | 1 |  | 1 | 0 | 0 |  | 0 | 0 | 1 |
| 1 | 1 | 1 |  | 0 | 0 | 1 |  | 1 | 0 | 0 |  | 1 | 1 | 1 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N | 1 | 0 | 1 |  | 1 | 1 | 1 |  |
| 1 | 1 | 1 |  | 0 | 1 | 0 |  |
| 1 | 0 | 1 |  | 1 | 1 | 1 |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| U | 1 | 0 | 1 |  | 1 | 1 | 1 |  | 1 | 1 | 1 |  | 1 | 1 | 1 |
| 1 | 0 | 1 |  | 1 | 0 | 1 |  | 1 | 0 | 0 |  | 0 | 0 | 1 |
| 1 | 1 | 1 |  | 1 | 0 | 1 |  | 1 | 1 | 1 |  | 1 | 1 | 1 |

You have a two-dimensional space containing n by m cells, where n and m are both greater than or equal to 3. Note that all n rows have the same number of columns, m.

Develop a program to count the occurrences of the letters L, N, and U in an n x m space, where n and m are greater than or equal to 3.

In cases where letters overlap are detected, each individual letter should be extracted and counted separately, as shown in the following example:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Col 1 | Col 2 | Col 3 | Col 4 | Col 5 | Col 6 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |

Here you have two L’s and one N

Col 1, Col 2 and Col 3 produce the first L

Col 2, Col 3 and Col 4 produce the second L

Col 4, Col 5 and Col 6 produce the only N

**Input data:** The data is stored in the input.txt file. The first line of the file indicates the number of rows (n), after it number of columns (m), followed by n rows of zeros and ones, separated by a single space.

Examples of different inputs:

3 3 The number of rows and columns the first case

1 0 0 The space where you need to look for the letters

1 0 0

1 1 1

3 8 The second test

1 1 1 1 1 1 0 1

1 0 0 0 0 1 1 1

1 0 0 1 1 1 0 1

4 6

1 1 1 1 1 1

1 1 0 1 1 1

0 0 1 0 1 1

0 0 0 0 0 0

3 6

1 0 0 1 0 1

1 0 0 1 1 1

1 1 1 1 0 1

6

6

**Output data:** The response must be written to the **output.txt** file, which include the occurrences of each letter. See the output of for the input examples above:

**Program execution time:** no more than 1 second.

**Note:** all characters in text files must be 1 byte in size.

**Examples:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test1:** | **Test2:** | **Test3:** | **Test4:** |
| **3 3**  **1 0 0**  **1 0 0**  **1 1 1** | **3 8**  **1 1 1 1 1 1 0 1**  **1 0 0 0 0 1 1 1**  **1 0 0 1 1 1 0 1** | **4 6**  **1 1 1 1 1 1**  **1 1 0 1 1 1**  **0 0 1 0 1 1**  **0 0 0 0 0 0** | **3 6**  **1 0 0 1 0 1**  **1 0 0 1 1 1**  **1 1 1 1 0 1** |
| **Answer1:** | **Answer2** | **Answer3** | **Answer4** |
| L-count 1  N-count 0  U-count 0 | L-count 1  N-count 1  U-count 1 | L-count 0  N-count 0  U-count 0 | L-count 2  N-count 1  U-count 0 |