Practice Arena

Practice problems aimed to improve your coding skills.

- PRACTICE-02_SCAN-PRINT
- PRACTICE-03_TYPES
- LAB-PRAC-02_SCAN-PRINT
- LAB-PRAC-01
- PRACTICE-04_COND
- BONUS-PRAC-02
- LAB-PRAC-03_TYPES
- PRACTICE-05_COND-LOOPS
- LAB-PRAC-04_COND
- LAB-PRAC-05_CONDLOOPS
- PRACTICE-07_LOOPS-ARR
- LAB-PRAC-06_LOOPS
- LAB-PRAC-07_LOOPS-ARR
- LABEXAM-PRAC-01_MIDSEM
- PRACTICE-09_PTR-MAT
- LAB-PRAC-08_ARR-STR
- PRACTICE-10_MAT-FUN
- LAB-PRAC-09_PTR-MAT
- LAB-PRAC-10_MAT-FUN
  - Stack
  - The Prutor Editor
  - Finding your identity
  - Queue
  - The Prutor Editor Part II
  - Only Ones
  - Graphs
  - How Mr C actually does Math
  - The Hidden Positives and Negatives
  - How Prutor Manages Memory
  - Message in the Matrix
  - The Hidden Key
- PRACTICE-11_FUN-PTR
- LAB-PRAC-11_FUN-PTR
- LAB-PRAC-12_FUN-STRUC
- LABEXAM-PRAC-02_ENDSEM
- LAB-PRAC-13_STRUC-NUM
- LAB-PRAC-14_SORT-MISC

Message in the Matrix

LAB-PRAC-10_MAT-FUN

Message in the Matrix [20 marks]

Problem Statement
In the first line you will be given two strictly positive integers m and n. In the next line you will given a string of length at most 100. The string will contain only lowercase English alphabets. In the next m lines, you will be given an m x n matrix of characters, with all elements being lowercase English alphabets, one row on each line. There will be no spaces between individual characters, nor any space at the end of any line.

You have to find whether the given string occurs in the matrix or not. The string can occur in the matrix in 3 ways - either left to right, or diagonally upwards, or diagonally downwards, as the example below suggests. Report all occurrences of the matrix by giving the row and column index of the first character of the occurrence. If there are multiple occurrences in the matrix, report them following the rules given below.

1. First, report all occurrences where the first character is in the first row, then all occurrences where the second row, and so on.
2. If there are multiple occurrences in a row, report them in increasing order of column index of the first character in the occurrence.
3. If there are multiple occurrences at a given index e.g. flat as well as diagonally up, then report the diagonally upward occurrence first (if present), then the flat occurrence (if present), then the diagonally downward occurrence first (if present).

Caution

1. Use scanf("%d %d\n", &m, &n); to read the size of the matrix so that you read in the newline at the end of the first line as well.
2. We will not penalize you for extra newlines at the end of your output. However, do not have extra newlines in the middle of your output or else have trailing spaces in any line of the output.
3. Use gets() to read the string. However, be careful if you are using getchar to read the matrix element by element since there will be newline characters at the end of each row which will need to be excluded.

EXAMPLE:
INPUT

```
3 3
ab
vbr
abq
cbd
```

OUTPUT:

```
FOUND AT (1, 0) DIAG UP
FOUND AT (1, 0) FLAT
FOUND AT (1, 0) DIAG DOWN
```

Grading Scheme:
Total marks: **[20 Points]**

There will be partial grading in this question. There are several lines in your output. Printing each line correctly, in the correct order, carries equal weightage. Each visible test case is worth 2 points and each hidden test case is worth 4 points. There are 2 visible and 4 hidden test cases.

Please remember, however, that when you press Submit/Evaluate, you will get a green bar only if all parts of your answer are correct. Thus, if your answer is only partly correct, Prutor will say that you have not passed that test case completely, but when we do autograding afterwards, you will get partial marks.

Start Solving! (/editor/practice/6207)