Sabancı University

Faculty of Engineering and Natural Sciences

CS204 Advanced Programming

Spring 2024

Homework1 – Tile Movement Game

Due: March 6, 2024, 21:00 (**not** midnight)

No Late Submissions

PLEASE NOTE:

Your program should be a robust one such that you have to consider all relevant programmer mistakes and extreme cases (with the exceptions of assumptions given in the homework text); you are expected to take actions accordingly!

You can NOT collaborate with your friends and discuss solutions. You have to write down the code on your own. Plagiarism will not be tolerated!

Introduction

The aim of this homework is to practice on the topic of two dimensional (2D) matrices implemented using `vector` of `vector`. You will use a **`vector`** of **`vector`** of **`char`** (i.e., **`char`** matrix) as the data structure to store and manipulate the data. The matrix content will basically be characters representing four main directions, and a character representing empty cell. The initial content of the matrix will be read from a text file. You will also read integer input from the standard input (keyboard). According to the input to be taken from the keyboard and the values on the matrix, your code will move the tiles in the cells toward its direction until it is blocked by another non-empty cell or drops off from the edge of the matrix. Additional information is given in the rest of this document.

Inputs

We have both file and keyboard input in the homework. File input is to read the source matrix. Keyboard inputs are for the row and column indices of the cells in the matrix (in order to move them while playing the game).

Input from file

A valid text file will contain a 2D matrix of characters structured in rows and columns. In each row, there should be the same number of characters, and there could be any number of rows. In the file, only five characters (four for directions, one to denote empty cell) can exist:

`r`	for right
`l`	for left
`u`	for up
`d`	for down
`-`	for empty cell

Figure 1 shows an example of such a sample input file content with 3 rows and 7 columns (indices are added to the figure as extra information; they are not part of the file content). Your program should work with all files having structures of any number of rows and any number of columns.

	0	**1**	**2**	**3**	**4**	**5**	**6**
0	r	-	u	u	d	u	r
1	l	u	-	u	-	-	l
2	-	u	d	r	d	u	-

Figure 1. Sample input file (indices are added for informative reasons; they are not part of the file content)

You should read the content of this file into a **`vector`** of **`vector`** of **`char`** data structure. You are **not** allowed to use another structure for this purpose.

We recommend you to read the file line-by-line using `getline` function, since it reads a line entirely into a string by automatically stripping off the invisible end-of-line character(s). After that, you can process the line string as you want. However, if you read by some other means, you take the risk of processing the invisible end-of-line characters; this might cause inconsistencies in different operating systems.

There are some input checks that you should perform for the file content.

- Your program should check for consistency of the number of characters in each row. In other words, all rows must be of the same length.
- Your program should check that the file only contains 4 lowercase letters and dash character as mentioned above. Any other letter/symbol (even blank character or blank lines) in a file will make that file invalid.

In case of a problem with the file content, your program must display a generic error message and quit. See the sample runs for the error message content.

Before reading the file content, you will need to read the file name and open it. There are some checks to be done with file opening/existence as well; this is going to be explained in the "Program Flow" section below.

Input from keyboard

After successfully reading the content from a valid input file and storing it in a **`vector`** of **`vector`** of **`char`** structure, your program will read multiple coordinates where each coordinate is in the form of *`row column`*, where *`row`* corresponds to the row (horizontal) index of a cell, and *`column`* corresponds to the column (vertical) index of a cell. The top left cell of the matrix has coordinates of (0,0) i.e. 0th *`row`* and 0th *`column`.*

If the user enters an invalid cell coordinates (i.e. out of indices or negative coordinates), your program should print a generic invalidity message indicating that, and keep asking for valid ones. Please see the sample runs for the content of the message and prompts.

You may assume that the user always enters integers for coordinates.

Program Flow

Your program will start by asking the user to enter a file name for the input text file. Then, your program will open the file and check if it is opened successfully or not. If, for some reason, the program could not open the file, it should keep reading another file name until opened successfully. When the program opens the file and reads its content into the matrix, firstly it should check for the validity of the matrix by checking the rules mentioned in the "Input" section above. If the matrix is not valid, then the program should display an error message and terminate.

On the other hand, if the matrix is a valid one, your program should print the matrix on the screen and then proceed with game playing. In each round of the game, the user will enter valid row and column indices of a cell as explained in the "Input from Keyboard" section above.

After that, your program should try to move the tile in that cell continuously in the direction denoted by that cell's content as long as it is not blocked by a non-empty cell preventing its movement. There are four different outcomes for the attempt to move.

1) If the position of the cell corresponds to the edge of corner of the matrix and the direction causes the cell to drop off, then the cell disappears and the position in the matrix is replaced by a dash character ('`-`') denoting that it is now empty.
2) If the path in the cell's direction is clear all the way (i.e. the cells are all empty cells until the edge of the matrix), then the cell drops off, meaning that it disappears. In this case, the original position will be replaced by a dash character ('`-`') denoting that it is now empty.
3) If the cell can move in its direction by one or more cells due to having empty cells on the path, but is blocked eventually by a non-empty cell, then the cell where it is blocked is replaced by the content of the original cell (to indicate the movement), and the original position will be replaced by a dash character ('`-`') denoting that it is now empty.

4) If no movement is possible due to having an immediate non-empty cell in the direction of the cell, then nothing changes in the matrix.

For the first three cases mentioned above, your program should display that there is a movement (see the sample runs for the structure of this message) and display the new content of the matrix. For the last case, your program should indicate the there is no movement (again see the sample runs for the message) and display the unchanged matrix again.

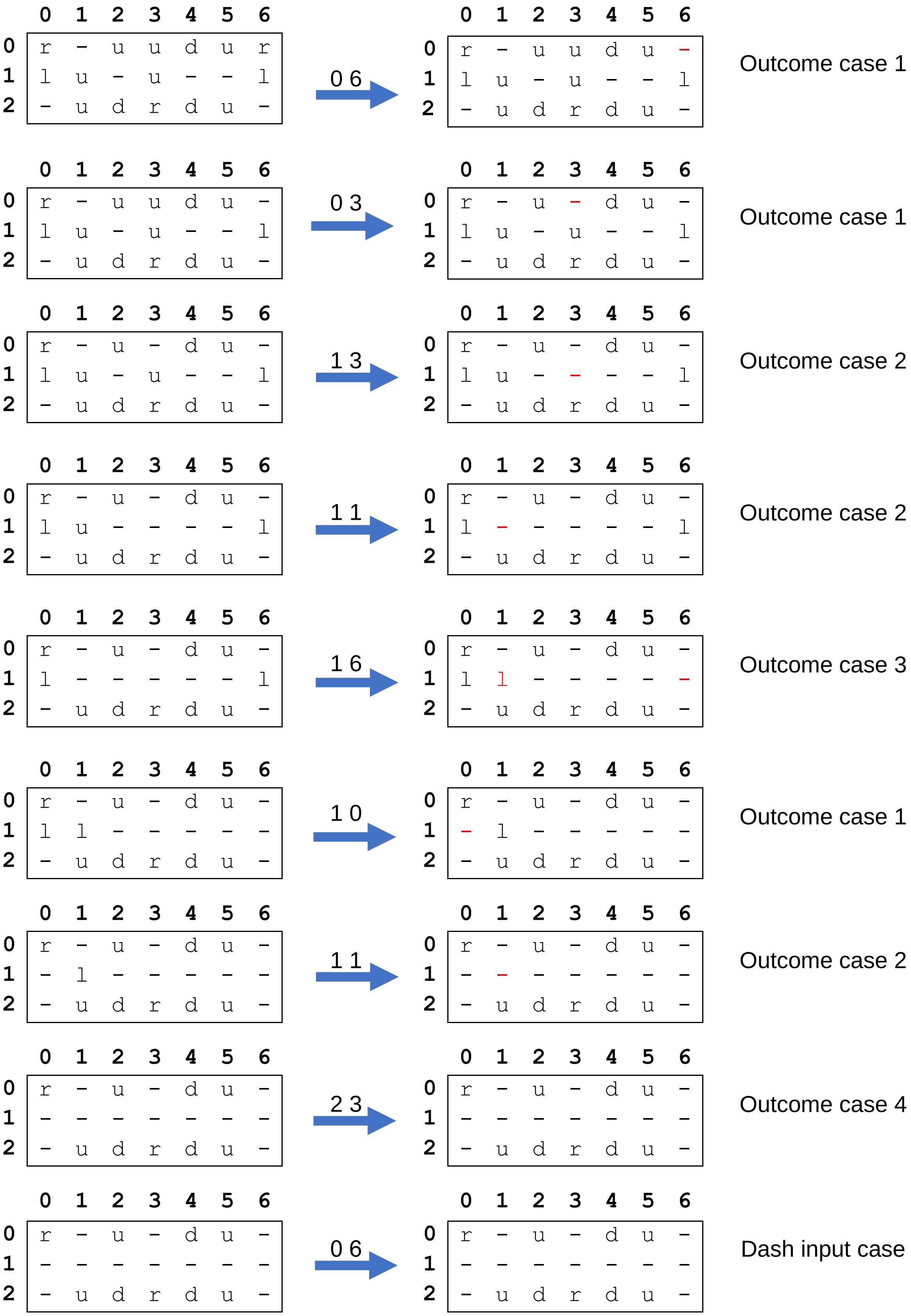
If the input coordinates correspond to an empty cell with dash character in it, then there will be no movement and the reaction of the program will be as in case 4 above.

Figure 2 below exemplifies these cases.

The game/program will terminate when one of the following two cases happens:

- The matrix became all empty (i.e. all cells contain dashes)
- There are no possible movements for any of the remaining non-empty cells.

When the program stops due to any of the above two reasons, your program should print a specific message indicating the reason for ending, and then quit. Please refer to sample runs for the exact expected output messages.



	0	1	2	3	4	5	6
0	r	-	u	u	d	u	r
1	l	u	-	u	-	-	l
2	-	u	d	r	d	u	-

0 6 →

	0	1	2	3	4	5	6
0	r	-	u	u	d	u	-
1	l	u	-	u	-	-	l
2	-	u	d	r	d	u	-

Outcome case 1

	0	1	2	3	4	5	6
0	r	-	u	u	d	u	-
1	l	u	-	u	-	-	l
2	-	u	d	r	d	u	-

0 3 →

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	l	u	-	u	-	-	l
2	-	u	d	r	d	u	-

Outcome case 1

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	l	u	-	u	-	-	l
2	-	u	d	r	d	u	-

1 3 →

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	l	u	-	-	-	-	l
2	-	u	d	r	d	u	-

Outcome case 2

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	l	u	-	-	-	-	l
2	-	u	d	r	d	u	-

1 1 →

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	l	-	-	-	-	-	l
2	-	u	d	r	d	u	-

Outcome case 2

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	l	-	-	-	-	-	l
2	-	u	d	r	d	u	-

1 6 →

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	l	l	-	-	-	-	-
2	-	u	d	r	d	u	-

Outcome case 3

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	l	l	-	-	-	-	-
2	-	u	d	r	d	u	-

1 0 →

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	-	l	-	-	-	-	-
2	-	u	d	r	d	u	-

Outcome case 1

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	-	l	-	-	-	-	-
2	-	u	d	r	d	u	-

1 1 →

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	-	-	-	-	-	-	-
2	-	u	d	r	d	u	-

Outcome case 2

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	-	-	-	-	-	-	-
2	-	u	d	r	d	u	-

2 3 →

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	-	-	-	-	-	-	-
2	-	u	d	r	d	u	-

Outcome case 4

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	-	-	-	-	-	-	-
2	-	u	d	r	d	u	-

0 6 →

	0	1	2	3	4	5	6
0	r	-	u	-	d	u	-
1	-	-	-	-	-	-	-
2	-	u	d	r	d	u	-

Dash input case

Figure 2. Some Example Movement Attempts

Some Important Programming Rules

Please do not use any non-ASCII characters (Turkish or other) in your code (not even as comments).

In order to get a full credit, your programs must be efficient and well presented, presence of any redundant computation or bad indentation, or missing, irrelevant comments are going to decrease your grades. You also have to use understandable identifier names, informative introduction and prompts. Modularity is also important; you have to use functions wherever needed and appropriate.

When we grade your homework, we pay attention to these issues. Moreover, in order to observe the real performance of your codes, we may run your programs in *Release* mode and **we may test your programs with very large test cases**. Of course, your program should work in *Debug* mode as well.

You are **not** allowed to use codes found somewhere online. These restrictions include code repositories, sites like geeksforgeeks, codes generated by GenAI tools, etc. Moreover, you are not allowed to use any statement, command, concept, topic that has not been covered in CS201 and CS204 (until now). Trying to find help online would generally cause such a problem. Thus, you always have to find help within the course material.

You cannot use `break` and `continue` statements. Global variables cannot be used either.

You are allowed to use sample codes shared with the class by the instructor and TAs. However, you cannot start with an existing .cpp or .h file directly and update it; you have start with an empty file. Only the necessary parts of the shared code files can be used and these parts must be clearly marked in your homework by putting comments like the following. Even if you take a piece of code and update it slightly, you have to put a similar marking (by adding "`and updated`" to the comments below.

```
/* Begin: code taken from lab1.cpp */
…
/* End: code taken from lab1.cpp */
```

Since CodeRunner configuration in this homework does not allow to use extra non-standard C++ library/function/class files, if you want to use such functions/classes covered in the classes (such as strutils), you will need to copy the necessary declarations and implementations to your main program by following the citation rules mentioned above.

Submission Rules and Some Grading Tips (PLEASE READ, IMPORTANT)

It'd be a good idea to write your name and last name in the program (as a comment line of course). Do not use any Turkish characters anywhere in your code (not even in comment parts). For example, if your full name is "Satılmış Özbugsızkodyazaroğlu", then you must type it as follows:

```
// Satilmis Ozbugsizkodyazaroglu
```

We use CodeRunner in SUCourse+ for submission. No other way of submission is possible. Since the functionality part of the grading process will be automatic, you have to strictly follow these guidelines; otherwise we cannot grade your homework.

The advantage CodeRunner it is that you will be able to test your code against sample test cases. However, the output should be exact, but the textual differences between the correct output and yours can be highlighted (by pressing "show differences" button) on the submission interface.

You should copy the full content of the main .cpp file and paste it into the specified “Answer” area in the relevant assignment submission page on SUCourse+. Then you can test your code via CodeRunner against the sample runs (by pressing the "Check" button). Since you will not upload a file, your local cpp file name is not important.

Even any tiny change in the output format will result in your grade being zero (0) for that particular test case, so please test your programs yourself, and against the sample runs that are available at the relevant assignment submission page on SUCourse+ (CodeRunner).

In the CodeRunner, there are some visible and invisible (hidden) test cases. You will know whether your code has successfully passed all the test cases or not before submitting your code. However, we keep our rights to add more test cases in the grading process after the submission. **Thus, please make sure that you have read this documentation carefully and covered/tested all possible cases, even some other cases you may not have seen on CodeRunner or the sample runs.** Due to these reasons, **your final grade may conflict with what you have seen on CodeRunner**. We will also **manually** check your code against some criteria, comments, indentations and so on; hence, please do not object to your grade based on the **CodeRunner** results, but rather, consider every detail on this documentation.

You have to manually "Submit" after you test and finish with your code (there is no automatic submission). There is no re-submission. That means, after you submit, you cannot take it back. On the other hand, this does not mean that you have one shot to test CodeRunner. You don't have to complete your task in one time, you can continue from where you left last time, but you should not press submit before finalizing it. Therefore, you should make sure that it’s your final solution version before you submit it. Also, we still do not suggest that you develop your solution on CodeRunner but rather on your IDE on your computer.

Last, even if you cannot completely finish your homework, you can still submit.

Please see the syllabus for general homework grading issues.

Plagiarism

Plagiarism is checked by automated tools, and we are very capable of detecting such cases. Be careful with that. Exchange of abstract ideas are totally okay but once you start sharing the code with each other, it is very probable to get caught by plagiarism. So, do NOT share any part of your code to your friends by any means or you might be charged as well, although you have done your homework by yourself.

Homework are to be done personally and you have to submit your own work. **Cooperation will NOT be counted as an excuse.**

Our experience shows that code taken from online sources or generated by AI tools also show resemblance; thus if you try to get such help, you also may be charged by plagiarism with a person that you do not know.

In case of plagiarism, the rules written in the Syllabus apply.

Sample Runs

Sample runs are given below, but these are not comprehensive, therefore you must consider **all possible cases** to get full mark. User inputs are shown in **bold.**

The input files of the visible test cases are also provided in the homework package. In C-Lion, input text files must be placed in the folder that starts with cmake-build-. This is something different than Visual Studio.

We configured CodeRunner to test these sample runs for you (as visible test cases). However, there also are some hidden test cases that would affect your grade. We will not disclose the hidden test cases before the grading has been completed.

We do **not** recommend you to copy and paste the prompts and messages from this document since some hidden control characters and non-standard characters might cause problems in CodeRunner.

Sample Run 1

```
Please enter the file name:
input.txt
Could not open the file. Please enter a valid file name:
myFile.txt
Could not open the file. Please enter a valid file name:
inp3
Could not open the file. Please enter a valid file name:
inp.text
Could not open the file. Please enter a valid file name:
inp3.txt
The matrix file contains:
---
---
The matrix is empty.
```

Sample Run 2

```
Please enter the file name:
inp7.txt
The matrix file contains:
d
u
No more moves are possible.
```

Sample Run 3

```
Please enter the file name:
inp2.txt
Erroneous file content. Program terminates.
```

Sample Run 4

```
Please enter the file name:
inp4.txt
Erroneous file content. Program terminates.
```

Sample Run 5

```
Please enter the file name:
inp6.txt
Erroneous file content. Program terminates.
```

Sample Run 6

```
Please enter the file name:
inp9.txt
The matrix file contains:
ludr--lud-udlr
Please enter the row and column of the tile that you want to move:
0   14
Invalid row or column index.
Please enter the row and column of the tile that you want to move:
0 13
Tile at (0,13) has been moved.
ludr--lud-udl-
Please enter the row and column of the tile that you want to move:
0  0
Tile at (0,0) has been moved.
-udr--lud-udl-
Please enter the row and column of the tile that you want to move:
0 3
Tile at (0,3) has been moved.
-ud--rlud-udl-
Please enter the row and column of the tile that you want to move:
0 2
Tile at (0,2) has been moved.
-u---rlud-udl-
Please enter the row and column of the tile that you want to move:
0 1
Tile at (0,1) has been moved.
-----rlud-udl-
Please enter the row and column of the tile that you want to move:
0         6
No tiles have been moved.
-----rlud-udl-
```

```
Please enter the row and column of the tile that you want to move:
0 7
Tile at (0,7) has been moved.
-----rl-d-udl-
Please enter the row and column of the tile that you want to move:
0 8
Tile at (0,8) has been moved.
-----rl---udl-
Please enter the row and column of the tile that you want to move:
0 9
No tiles have been moved.
-----rl---udl-
Please enter the row and column of the tile that you want to move:
0 10
Tile at (0,10) has been moved.
-----rl----dl-
Please enter the row and column of the tile that you want to move:
0 12
No tiles have been moved.
-----rl----dl-
Please enter the row and column of the tile that you want to move:
0 13
No tiles have been moved.
-----rl----dl-
Please enter the row and column of the tile that you want to move:
0 11
Tile at (0,11) has been moved.
-----rl-----l-
Please enter the row and column of the tile that you want to move:
0 14
Invalid row or column index.
Please enter the row and column of the tile that you want to move:
0 15
Invalid row or column index.
Please enter the row and column of the tile that you want to move:
0 13
No tiles have been moved.
-----rl-----l-
Please enter the row and column of the tile that you want to move:
0 12
Tile at (0,12) has been moved.
-----rll------
No more moves are possible.
```

Sample Run 7

```
Please enter the file name:
inp5.txt
The matrix file contains:
lrrurrr
lrdduur
uldlrrr
Please enter the row and column of the tile that you want to move:
0 2
No tiles have been moved.
lrrurrr
lrdduur
uldlrrr
Please enter the row and column of the tile that you want to move:
2 0
No tiles have been moved.
lrrurrr
lrdduur
uldlrrr
Please enter the row and column of the tile that you want to move:
2 6
Tile at (2,6) has been moved.
lrrurrr
lrdduur
uldlrr-
Please enter the row and column of the tile that you want to move:
2 4
No tiles have been moved.
lrrurrr
lrdduur
uldlrr-
Please enter the row and column of the tile that you want to move:
2 5
Tile at (2,5) has been moved.
lrrurrr
lrdduur
uldlr--
Please enter the row and column of the tile that you want to move:
2 2
Tile at (2,2) has been moved.
lrrurrr
lrdduur
ul-lr--
Please enter the row and column of the tile that you want to move:
0 0
Tile at (0,0) has been moved.
-rrurrr
```

```
lrdduur
ul-lr--
Please enter the row and column of the tile that you want to move:
0 1
No tiles have been moved.
-rrurrr
lrdduur
ul-lr--
Please enter the row and column of the tile that you want to move:
1 0
Tile at (1,0) has been moved.
-rrurrr
-rdduur
ul-lr--
Please enter the row and column of the tile that you want to move:
2 0
Tile at (2,0) has been moved.
-rrurrr
-rdduur
-l-lr--
Please enter the row and column of the tile that you want to move:
0 2
No tiles have been moved.
-rrurrr
-rdduur
-l-lr--
Please enter the row and column of the tile that you want to move:
0 3
Tile at (0,3) has been moved.
-rr-rrr
-rdduur
-l-lr--
Please enter the row and column of the tile that you want to move:
1 3
No tiles have been moved.
-rr-rrr
-rdduur
-l-lr--
Please enter the row and column of the tile that you want to move:
1 5
No tiles have been moved.
-rr-rrr
-rdduur
-l-lr--
Please enter the row and column of the tile that you want to move:
1 4
No tiles have been moved.
-rr-rrr
```

```
-rdduur
-l-lr--
Please enter the row and column of the tile that you want to move:
2 3
Tile at (2,3) has been moved.
-rr-rrr
-rdduur
-ll-r--
Please enter the row and column of the tile that you want to move:
2 2
No tiles have been moved.
-rr-rrr
-rdduur
-ll-r--
Please enter the row and column of the tile that you want to move:
0 1
No tiles have been moved.
-rr-rrr
-rdduur
-ll-r--
Please enter the row and column of the tile that you want to move:
0 8
Invalid row or column index.
Please enter the row and column of the tile that you want to move:
0 7
Invalid row or column index.
Please enter the row and column of the tile that you want to move:
0 6
Tile at (0,6) has been moved.
-rr-rr-
-rdduur
-ll-r--
Please enter the row and column of the tile that you want to move:
0 5
Tile at (0,5) has been moved.
-rr-r--
-rdduur
-ll-r--
Please enter the row and column of the tile that you want to move:
0 4
Tile at (0,4) has been moved.
-rr----
-rdduur
-ll-r--
Please enter the row and column of the tile that you want to move:
0 2
Tile at (0,2) has been moved.
-r-----
```

```
-rdduur
-ll-r--
Please enter the row and column of the tile that you want to move:
0 1
Tile at (0,1) has been moved.
-------
-rdduur
-ll-r--
Please enter the row and column of the tile that you want to move:
1 4
Tile at (1,4) has been moved.
-------
-rdd-ur
-ll-r--
Please enter the row and column of the tile that you want to move:
1 5
Tile at (1,5) has been moved.
-------
-rdd--r
-ll-r--
Please enter the row and column of the tile that you want to move:
1 6
Tile at (1,6) has been moved.
-------
-rdd---
-ll-r--
Please enter the row and column of the tile that you want to move:
1 3
Tile at (1,3) has been moved.
-------
-rd----
-ll-r--
Please enter the row and column of the tile that you want to move:
2 1
Tile at (2,1) has been moved.
-------
-rd----
--l-r--
Please enter the row and column of the tile that you want to move:
2 2
Tile at (2,2) has been moved.
-------
-rd----
----r--
Please enter the row and column of the tile that you want to move:
1 1
No tiles have been moved.
-------
```

```
-rd----
----r--
Please enter the row and column of the tile that you want to move:
1 2
Tile at (1,2) has been moved.
-------
-r-----
----r--
Please enter the row and column of the tile that you want to move:
1 1
Tile at (1,1) has been moved.
-------
-------
----r--
Please enter the row and column of the tile that you want to move:
2 3
No tiles have been moved.
-------
-------
----r--
Please enter the row and column of the tile that you want to move:
2 4
Tile at (2,4) has been moved.
-------
-------
-------
The matrix is empty.
```

Sample Run 8

```
Please enter the file name:
inp1.txt
The matrix file contains:
r-uudur
lu-u--l
-udrdu-
Please enter the row and column of the tile that you want to move:
0 5
Tile at (0,5) has been moved.
r-uud-r
lu-u--l
-udrdu-
Please enter the row and column of the tile that you want to move:
0 2
Tile at (0,2) has been moved.
r--ud-r
lu-u--l
-udrdu-
Please enter the row and column of the tile that you want to move:
```

```
0 3
Tile at (0,3) has been moved.
r---d-r
lu-u--l
-udrdu-
Please enter the row and column of the tile that you want to move:
2 1
No tiles have been moved.
r---d-r
lu-u--l
-udrdu-
Please enter the row and column of the tile that you want to move:
1 1
Tile at (1,1) has been moved.
r---d-r
l--u--l
-udrdu-
Please enter the row and column of the tile that you want to move:
2 1
Tile at (2,1) has been moved.
r---d-r
l--u--l
--drdu-
Please enter the row and column of the tile that you want to move:
0 4
Tile at (0,4) has been moved.
r-----r
l--ud-l
--drdu-
Please enter the row and column of the tile that you want to move:
1 4
No tiles have been moved.
r-----r
l--ud-l
--drdu-
Please enter the row and column of the tile that you want to move:
2 4
Tile at (2,4) has been moved.
r-----r
l--ud-l
--dr-u-
Please enter the row and column of the tile that you want to move:
1 4
Tile at (1,4) has been moved.
r-----r
l--u--l
--dr-u-
Please enter the row and column of the tile that you want to move:
```

```
0 0
Tile at (0,0) has been moved.
-----rr
l--u--l
--dr-u-
Please enter the row and column of the tile that you want to move:
0 7
Invalid row or column index.
Please enter the row and column of the tile that you want to move:
0 5
No tiles have been moved.
-----rr
l--u--l
--dr-u-
Please enter the row and column of the tile that you want to move:
0 6
Tile at (0,6) has been moved.
-----r-
l--u--l
--dr-u-
Please enter the row and column of the tile that you want to move:
0 5
Tile at (0,5) has been moved.
-------
l--u--l
--dr-u-
Please enter the row and column of the tile that you want to move:
1 6
Tile at (1,6) has been moved.
-------
l--ul--
--dr-u-
Please enter the row and column of the tile that you want to move:
1 3
Tile at (1,3) has been moved.
-------
l---l--
--dr-u-
Please enter the row and column of the tile that you want to move:
1 4
Tile at (1,4) has been moved.
-------
ll-----
--dr-u-
Please enter the row and column of the tile that you want to move:
1 1
No tiles have been moved.
-------
```

```
ll-----
--dr-u-
Please enter the row and column of the tile that you want to move:
1 0
Tile at (1,0) has been moved.
-------
-l-----
--dr-u-
Please enter the row and column of the tile that you want to move:
1 1
Tile at (1,1) has been moved.
-------
-------
--dr-u-
Please enter the row and column of the tile that you want to move:
2 3
Tile at (2,3) has been moved.
-------
-------
--d-ru-
Please enter the row and column of the tile that you want to move:
2 4
No tiles have been moved.
-------
-------
--d-ru-
Please enter the row and column of the tile that you want to move:
2 5
Tile at (2,5) has been moved.
-------
-------
--d-r--
Please enter the row and column of the tile that you want to move:
2 4
Tile at (2,4) has been moved.
-------
-------
--d----
Please enter the row and column of the tile that you want to move:
2 2
Tile at (2,2) has been moved.
-------
-------
-------
The matrix is empty.
```

Sample Run 9

```
Please enter the file name:
inp10.txt
The matrix file contains:
---du-l---
----d-----
--d---r--d
--u-------
----------
Please enter the row and column of the tile that you want to move:
0 2
No tiles have been moved.
---du-l---
----d-----
--d---r--d
--u-------
----------
Please enter the row and column of the tile that you want to move:
0 3
Tile at (0,3) has been moved.
----u-l---
----d-----
--d---r--d
--u-------
----------
Please enter the row and column of the tile that you want to move:
0 6
Tile at (0,6) has been moved.
----ul----
----d-----
--d---r--d
--u-------
----------
Please enter the row and column of the tile that you want to move:
0 4
Tile at (0,4) has been moved.
-----l----
----d-----
--d---r--d
--u-------
----------
Please enter the row and column of the tile that you want to move:
0 5
Tile at (0,5) has been moved.
----------
----d-----
--d---r--d
--u-------
```

```
----------
Please enter the row and column of the tile that you want to move:
2 2
No tiles have been moved.
----------
----d-----
--d---r--d
--u-------
----------
Please enter the row and column of the tile that you want to move:
3 2
No tiles have been moved.
----------
----d-----
--d---r--d
--u-------
----------
Please enter the row and column of the tile that you want to move:
1 4
Tile at (1,4) has been moved.
----------
----------
--d---r--d
--u-------
----------
Please enter the row and column of the tile that you want to move:
2 6
Tile at (2,6) has been moved.
----------
----------
--d-----rd
--u-------
----------
Please enter the row and column of the tile that you want to move:
2 7
No tiles have been moved.
----------
----------
--d-----rd
--u-------
----------
Please enter the row and column of the tile that you want to move:
2 8
No tiles have been moved.
----------
----------
--d-----rd
--u-------
```

```
----------
Please enter the row and column of the tile that you want to move:
2 9
Tile at (2,9) has been moved.
----------
----------
--d-----r-
--u-------
----------
Please enter the row and column of the tile that you want to move:
2 8
Tile at (2,8) has been moved.
----------
----------
--d-------
--u-------
----------
No more moves are possible.
```

Good Luck
Ahmed Salem and Albert Levi