Week 10 Graded Problem - Quiz 5

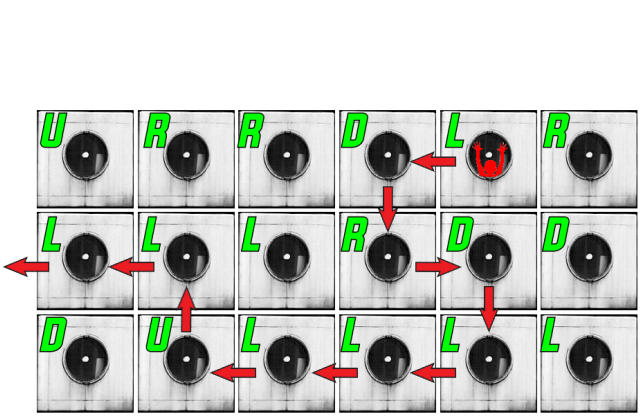
Topic: Graph Traversal

Built in 1972, the Nakagin Capsule Tower was considered as Japan's most prominent symbol of metabolism architecture. However, years of disrepair and decay have forced the owners to bring down the structure. The capsules have been removed one by one by a construction waste disposing company to be demolished. However, something went wrong. The clerk in charge of approving the legal documentation was a psychopath who saw a great opportunity in this. He changed the address where the capsules were to be delivered for demolition to a faraway abandoned place, where he could carry out his evil plan. His idea is to use the capsules to build a compound that would trap people with little or no hope of escape, so he could watch their despair as they try to get free.

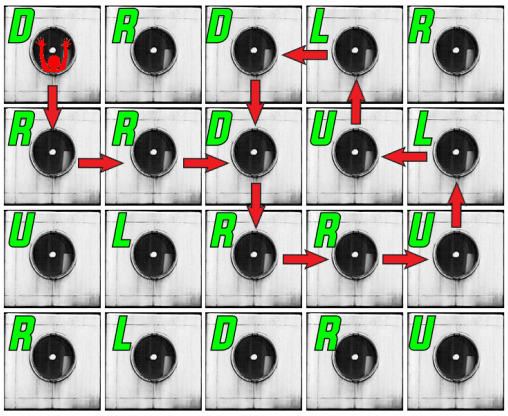
He stacks the capsules next to and on top of each other to create a compound of several **floors** and **columns**, making sure each capsule has only one way in and one way out (the entrance and exit may or may not be on the same wall). The entrapped person can move between capsules, but they must follow the instructions in each capsule that leads them over a predetermined path through the compound. The instructions specify which way each capsule can be exited, and each capsule can be exited in one direction only. A capsule's one-way exit leads either into another capsule or out of the compound into freedom. The possible instructions (movement directions) are:

```
U - Up (up the compound)
D - Down (down the compound)
R - Right (to the right inside the compound)
L - Left (to the left inside the compound)
```

The entrapped person will always be dropped in one of the capsules at the top floor, from where they will start their terrifying path towards either their freedom or entrapped demise. For example, suppose the compound is designed as **Compound 1** in the figure below. If the entrapped person is dropped in the top capsule in column 5, the first move they can make is to go left into the adjacent capsule because the instruction is to go LEFT. The entire path the entrapped person follows is shown in the figure. The person takes 10 steps inside the compound before they can leave it and become free.



COMPOUND 1



COMPOUND 2

Let's see what happens in **Compound 2**: If the entrapped person is dropped in the top capsule in column 1, they can make 3 steps, and then they get entrapped in a loop of 8 steps from which they can never exit.

Input

The input file name will be given as the first command line argument to your Main class in the following format:

```
java Main <inputfile>
```

The input will be given in a *.dat* file that contains data about the compound and the starting position. Three integers separated by blanks (spaces) will be given in the first line: **(1)** the *number of floors* in the compound, **(2)** the *number of columns* in the compound, and **(3)** the *number of the starting column* in which the entrapped person is dropped. The columns are numbered from 1 starting from the left. The remaining rows in the input file represent the direction instructions for each capsule. **Each compound will have at least one and at most 10 floors and columns**. The lines with instructions contain only the characters U, D, R, or L with no blanks. E.g., for Compound 1 with 3 floors and 6 columns the **input file** will be:

```
3 6 5
URRDLR
LLLRDD
DULLLL
```

For a second sample input for Compound 2 with 4 floors and 5 columns, where the entrapped person is dropped in column 1, the **input file** will be:

```
4 5 1
DRDLR
RRDUL
ULRRU
RLDRU
```

Output

You are expected to write a program that outputs the information about the number of steps taken either to freedom or when the person gets entrapped, to the STDOUT, in the formats given below.
For the given sample Compound 1, your output should be:

```
10 step(s) to freedom. Yay!
```

And for the given sample Compound 2 in which the person stays entrapped, your output should be:

```
3 step(s) before getting stuck in a loop of 8 step(s).
```

Important Rules and Grading Policy

- **You MUST use this starter code. Do not change any function signatures or member variables in the given starter codes, those are given to ensure that you pass the unit tests in autograding. Only complete the TODOs. You may add extra functions and variables if you wish.**
- **TODOs and Grading policy**:

  Implement **main** function by completing TODOs in *Main.java* such that variables **hasLoop**, **totalSteps**, and **loopSteps** will get assigned correct values:

  - Correct calculation of the total steps: 35%.
  - Correct detection of an existing loop: 20%.
  - Correct calculation of the loop steps: 35%.
  - Output tests: 10%.

- Test your codes using the autograding platform and finally submit them via submit.cs.hacettepe.edu.tr using the same format given below:
  - **<studentID>.zip**
    - **Main.java**