Assignment 3 Radegast's Game

Due: May 10th, 2023 (04.00 AM) CMPE 160

osman.bastug@boun.edu.tr , er.ilgaz@gmail.com, ece.ada@boun.edu.tr

INTRODUCTION

I am Radegast, the mighty money god of 6-figure salary-earning programmers. I possess an abundance of wealth that exceeds your wildest dreams. However, I feel bored and sad these days because I can predict the rise of AGI, resulting in my servants losing their 9-5 dream jobs :/ Hence, I want to play a game with you to test your programming skills and love for me. In the end, I will award you with all the money that you have successfully obtained.

The game consists of modifying a floating terrain by adding blocks of dirt at specific coordinates in the command line. You will have 10 opportunities to make modifications, after which I will flood the terrain with money and present it to you. All the money on it is yours if you can manage to carry out my commands below successfully.

1. Details

You have received a 2D positive integer matrix representing a terrain map. Each integer in the matrix signifies the height of the terrain at that specific coordinate. You first need to reprint this terrain's height table with algebraic notation. This means that integers will start from 0 on the left and alphabetic strings at the bottom. We will specify the notation below.

Once you have the terrain map with algebraic notation, you will modify the table 10 times by incrementing the height of the coordinate given at the command line. First, you need to ask the coordinate as "Add stone i / 10 to coordinate:" where "i" refers to the modification number (1 for the first modification). After taking each coordinate and implementing the modification, you need to print the table again.

After all the modifications are complete, the god Radegast will flood the terrain with money. Your task is to find the money lakes that form in the holes shaped by you (10 modifications) and Radegast (initial terrain). You need to label the connected money puddles with the same alphabetic string notation, which we will specify below. After labeling, you again need to print the terrain but with the labeled money lakes.

Finally, Radegast requires you to calculate the volumes of the labeled money lakes, take their square roots, and print their sum.

2. Input & Output

Input begins with a line containing two space-separated integers M and N. Then N lines of M integers follow. These integers imply the initial heights of this terrain. An example input is provided below:

```
7 7
3 3 3 3 3 3 3
3 1 2 3 1 2 3
3 3 3 1 3 3 3
3 3 3 3 3 3 3
3 1 2 3 1 2 3
3 3 3 3 3 3 3
3 3 3 3 3 3 2
```

This is a 7 x 7 terrain.

Once the program reads the input from the "input.txt" file, the program should print the table and prompt the user to input the coordinates for modifications.

```
 0  3  3  3  3  3  3  3
 1  3  1  2  3  1  2  3
 2  3  3  3  1  3  3  3
 3  3  3  3  3  3  3  3
 4  3  1  2  3  1  2  3
 5  3  3  3  3  3  3  3
 6  3  3  3  3  3  3  2
    a  b  c  d  e  f  g
Add stone 1 / 10 to coordinate:
```

After printing the table and question provided above, you should give an algebraic coordinate input in the command line like “a2”, “d3” or “g5” and print the current table afterward. You should repeat this 10 times.

For example, if you give an input “d3” then the game should print the modified table and the next modification question as follows:

```
>>> d3
 0  3  3  3  3  3  3  3
 1  3  1  2  3  1  2  3
 2  3  3  3  1  3  3  3
 3  3  3  3  4  3  3  3
 4  3  1  2  3  1  2  3
 5  3  3  3  3  3  3  3
 6  3  3  3  3  3  3  2
    a  b  c  d  e  f  g
---------------
Add stone 2 / 10 to coordinate:
```

After 10 modifications, last height table,‘`---------------`’, labeled height table and the final score (with 2 significant figures) should be printed as follows:

```
 0  3  3  3  3  3  3  3
 1  3  A  A  3  A  A  3
 2  3  3  3  A  3  3  3
 3  3  3  3  4  4  4  4
 4  3  B  B  4  C  C  4
 5  3  3  3  4  4  4  4
 6  3  3  3  3  3  3  2
    a  b  c  d  e  f  g
Final score: 6.80
```

Labels ‘A’, ‘B’ and ‘C’ denote different the money lakes in alphabetical notation. The name of the lakes will be labeled (in order of precedence) from left to right, then top to bottom in ascending order: A, B, C .... AA, AB ... AZ, BA, BB, BC ... ZX, ZY, ZZ where adjacent lakes have the same labels.

The score is computed using $Score = \sum_{i} \sqrt{volume_{i}}$

where ***i*** and **Volume**$_i$ denote the label of the lake$_i$, and the volume of the money lake$_i$ respectively.

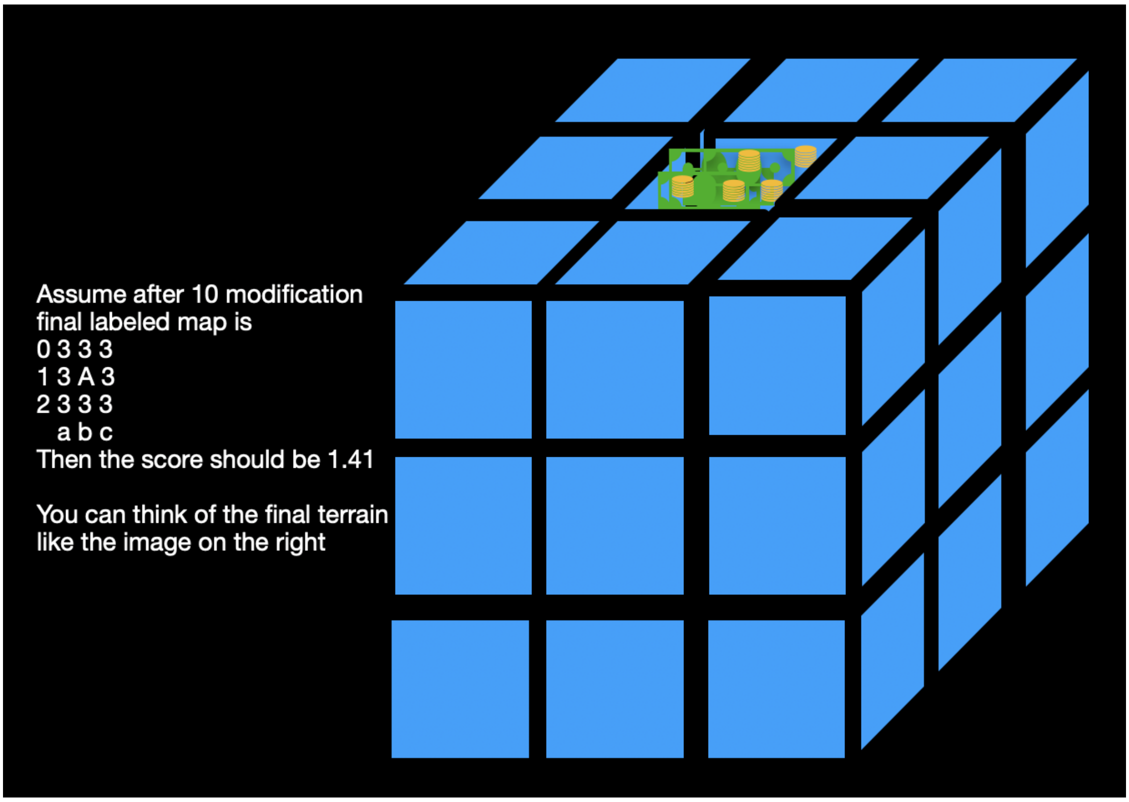
In the example_output.txt, since lake A has a volume of 8, B has a volume of 3, and C has a volume of 5. Hence the score is √8 +√3+ √5 ≅ 6.80

INPUT & OUTPUT FORMATS

`alphabetical notations at the bottom of the height table`	> start alphabetically like a, b, c. After 26 columns, column notations continue as follows: aa, ab … az, ba, bb, bc …zx, zy, zz >row or column numbers will not exceed 26*27
`alphabetical notations for the labeled height of the money lakes`	>starts alphabetically like A, B, C. After 26 columns, column notations continue as follows: AA, AB … AZ, BA, BB, BC … ZX, ZY, ZZ >number of money lakes will not exceed 26*27
`movement list`	>10 lines of algebraic notation which contains the alphabetic notation and the index of the line like "b7" or "aa13". Just like a chessboard! >if the input does not fit this notation, you should print: "Not a valid step!" and print "Add stone x / 10 to coordinate:" again where 'x' is the step's number.

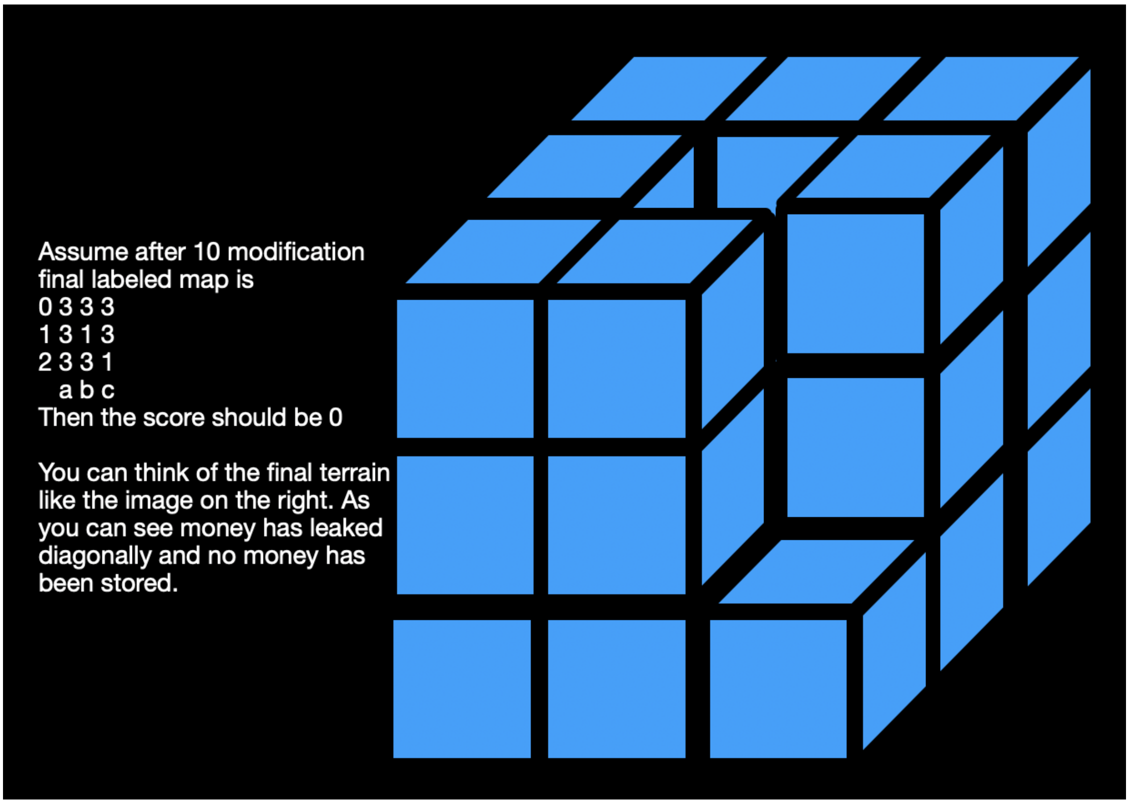
Important Information:

- Money is immaculately liquid. It can leak diagonally, and two pieces of water diagonal from each other belong to the same lake.
- Since the terrain is floating, the money cannot be stored outside of the terrain.
- Your goal is to create Radegast's game. We will test your code using the 3 input files (input.txt, input2.txt, and input3.txt) provided in the assignment directory and 7 hidden input files.
- The list after the corresponding terrain input data are the user input that will be used for testing. The resulting command line output should **exactly** match the output files (example_output.txt, example_output2.txt, example_output3.txt) provided in the assignment directory, or they will fail that test case.
- You should read the preexisting terrain from a txt file and print your outputs to the command line interface. Do not write your output to a .txt file or otherwise create any files. Creating any files will immediately fail all test cases.
- In your report, please include
  - the explanation of the algorithm (including the algorithm used for lake identification) without code (max 1 page)
  - the input and the output of all the example cases provided above.
- The visualizations of 2 different cases are provided below:



Assume after 10 modification
final labeled map is
0 3 3 3
1 3 A 3
2 3 3 3
  a b c
Then the score should be 1.41

You can think of the final terrain
like the image on the right



Assume after 10 modification
final labeled map is
0 3 3 3
1 3 1 3
2 3 3 1
  a b c
Then the score should be 0

You can think of the final terrain
like the image on the right. As
you can see money has leaked
diagonally and no money has
been stored.

Evaluation Criteria and Grading for Assignments

Code

90% Correctness of the solution and Object-Oriented Design Choices, programming best practices,compliance with programming style, e.g., naming conventions, indentation, and comments.

Report

10% Completeness of the report, compliance with the report format, and correctness of the content and language.

Submission Guide

Submission Files

Submit a single compressed (.zip) file to Moodle.
Name your zip file as name_surname.zip.
The zip file should contain all source codes (under the \code directory), and the report (in PDF format, under the \report directory).
Name the main code which is used to run your assignment as name_surname.java.
Name your report as name_surname.pdf.
The contents of each Java file should start with your name, student ID, date, and a brief code summary in a comment block.
DO NOT SAVE ANYTHING IN THE SUBMITTED CODE: Please refrain from saving or modifying any files within the submitted code, as it may affect the accuracy and reliability of our testing procedures.
DO NOT RENAME THE INPUT FILES: Please keep the original names of the input files as provided in the requirements.
FOLLOW THE SAME INPUT-OUTPUT FORMAT: It is crucial to adhere to the input-output format specified in the requirements for consistency and compatibility with our testing procedures. Do not use "Please enter station name:"
INCLUDE COMMENTS IN THE CODE: It is mandatory to provide comments in the code describing each method used. Failure to do so may result in a deduction of points.
WRITE THE REPORTS AND COMMENTS IN ENGLISH: Please ensure that all reports and comments are written in English, as reports in Turkish will not be accepted.

Mandatory Submission

Submission of assignments is mandatory. If you do not submit an assignment, you will fail the course.

Late Submission Policy

The maximum submission delay is two days. Late submissions will be graded on a scale of 50% of the original grade. Submission is mandatory even if you submit your assignment late.

Plagiarism

Plagiarism leads to grade F, and YÖK regulations will be applied