

CmpE 150 Introduction to Computing, Fall 2018
Project 3 - Due:23/12/2018, 23:59

Summary: In this homework you will

- read a table (label&digit pairs) from a file,
- print the table (only the digits) to the console,
- process the table according to the given rules,
- print the table (only the digits) to the console.

Submission: You will submit a project report and your code over Moodle. Project report should consist of the following sections:

1. State if you choose option 1 or option 2 (see below).
2. State which labels you can handle (see below). e.g. F, R and D.
3. This section will include your whole code with comments. You need to pay attention to indentation in order to improve readability.
 ↘ You do not need to explain each variable. Shortly explain each method before method definition.
4. Output of the program: A screen-shot of your program output should be put in this section. Two example runs are enough.

Your .java file should be named with your initials and your student number together (e.g., OS2013800027). If you have Turkish characters as your initials, please change them to non-Turkish. (Example: ÖS2013800027 should be OS2013800027). You will submit these over Moodle as a single zip file where the file name is your student number. Your zip file should consist of your .java file and your report in .doc or .pdf format. Do not use any Turkish characters in your code, class/variable names, or .java file names.

See the partial and late submission policy below in grading section.

Grading: Your code is automatically graded as usual. You can get partial credits as follows:

- Runs for inputs with only F's: +10
- Runs for inputs with only F's and R's: +15
- Runs for inputs with only F's and C's: +15
- Runs for inputs with only F's and D's: +20
- Runs for inputs with only F's and N's: +20
- Runs for inputs with all mixed: +20

The final grade is calculated as follows:

- Option A: With no for or while in your code:
 ↘ Final grade = (partial sum – lateDays x 10) x 1.0
- Option B: With for/while's in your code:
 ↘ Final grade = (partial sum – lateDays x 10) x 0.7

Important:

- You cannot use static class variables.
- You cannot use any method or approach that was not covered in the class, e.g. lambda expressions, methods as arguments of other methods, etc.
- If you select option A, do not include the keywords for or while in your code, including the comments.

Input-Output: Tables might be in different sizes, i.e. with different number of rows and columns. The input file is named "input.txt". An example input file for an example table of 4x5 size is as follows:

```
4x5
F1 N4 R2 F8 F1
F1 N2 N1 N6 N0
N1 D2 C1 N2 N2
N3 F1 N7 N1 R1
```

where

- 4 is the row number
- 5 is the column number
- F stands for Fixed label
- R stands for Row label.
- D stands for Diagonal label.
- C stands for Column label.
- N stands for Neighbor label.
- Each number will be a single digit.

After the table is read from the file, your program should print out the numbers to the console first:

```
1 4 2 8 1
1 2 1 6 0
1 2 1 2 2
3 1 7 1 1
```

Next, it should process each cell one-by-one:

- Processing starts from the left-most cell in the first row,
- After all the cells are processed in the first row, continue from the left-most digit in the second row.
- Processing continues until end.

After processing all the cells, you will output the latest state of the table to the console:

```
1 4 8 8 1
1 4 4 4 4
1 3 4 4 4
1 1 4 4 4
```

Descriptions of the labels and processing rules:

- F stands for Fixed. The values of the F cells should be kept constant. (Should not change.)
- R stands for Row. The values of the R cells should be replaced by the maximum number in the corresponding row.
- C stands for Column. The values of the C cells should be replaced by the median (not mean or average!) of the numbers in the corresponding row. In case there are even number of cells in the corresponding column, the smaller value should be used.
- D stands for Diagonal. The values of the D cells should be replaced by the mean of the numbers that lie in both diagonal directions (including the cell itself). In case the mean is not an integer, the closest smallest integer should be used instead.
- N stands for Neighbor. The values of the neighbors of the N cells and the neighbors of the neighbors should be replaced with the value of the processed N cell, recursively, only if those neighbors are also labeled as N. Only consider 4-cell neighborhood, i.e. the cells at the diagonals are not considered immediate neighbor.

An example trace of processing cells one by one: This is just an example of the trace, you should not output these intermediate steps.

// Initial state of the table:

F1 N4 R2 F8 F1
F1 N2 N1 N6 N0
N1 D2 C1 N2 N2
N3 F1 N7 N1 R1

// F cell value never changes.

// State of the table after processing (0,0)

F1 N4 R2 F8 F1
F1 N2 N1 N6 N0
N1 D2 C1 N2 N2
N3 F1 N7 N1 R1

// N cell sets the (N)ighbor cell values to 4, recursively.

// State of the table after processing (0,1)

F1 **N4** R2 F8 F1
F1 **N4 N4 N4 N4**
N1 D2 C1 **N4 N4**
N3 F1 **N4 N4** R1

// R takes the maximum value in its row.

// State of the table after processing (0,2)

F1 N4 R8 F8 F1
F1 N4 N4 N4 N4
N1 D2 C1 N4 N4
N3 F1 N4 N4 R1

//The next states:

F1 N4 R8 **F8** F1
F1 N4 N4 N4 N4
N1 D2 C1 N4 N4
N3 F1 N4 N4 R1

F1 N4 R8 F8 **F1**
F1 N4 N4 N4 N4
N1 D2 C1 N4 N4
N3 F1 N4 N4 R1

F1 N4 R8 F8 F1
F1 N4 N4 N4 N4
N1 D2 C1 N4 N4
N3 F1 N4 N4 R1

F1 N4 R8 F8 F1
F1 **N4 N4 N4 N4**
N1 D2 C1 **N4 N4**
N3 F1 **N4 N4** R1

F1 N4 R8 F8 F1
F1 **N4 N4 N4 N4**
N1 D2 C1 **N4 N4**
N3 F1 **N4 N4** R1

F1 N4 R8 F8 F1
F1 **N4 N4 N4 N4**
N1 D2 C1 **N4 N4**
N3 F1 **N4 N4** R1

F1 N4 R8 F8 F1
F1 **N4 N4 N4 N4**
N1 D2 C1 **N4 N4**
N3 F1 **N4 N4** R1

// N cell sets the (N)eighbor cell values to 1, recursively.

F1 N4 R8 F8 F1
F1 N4 N4 N4 N4
N1 D2 C1 N4 N4
N1 F1 N4 N4 R1

// D takes the mean of the values in the diagonal (including itself): $(1+2+4+1+4+8)/6 = 20/6 = 3$

F1 N4 R8 **F8** F1
F1 N4 **N4** N4 N4
N1 **D3** C1 N4 N4
N1 F1 **N4** N4 R1

// C takes the median of the values in its row: median of (1,4,4,8) is 4

F1 N4 **R8** F8 F1
F1 N4 **N4** N4 N4
N1 D3 **C4** N4 N4
N1 F1 **N4** N4 R1

F1 N4 R8 F8 F1
F1 N4 N4 N4 N4
N1 D3 C4 **N4 N4**
N1 F1 N4 N4 R1

F1 N4 R8 F8 F1
F1 N4 N4 N4 N4
N1 D3 C4 **N4 N4**
N1 F1 N4 N4 R1

F1 N4 R8 F8 F1
F1 N4 N4 N4 N4
N1 D3 C4 N4 N4
N1 F1 N4 N4 R1

F1 N4 R8 F8 F1
F1 N4 N4 N4 N4
N1 D3 C4 N4 N4
N1 **F1** N4 N4 R1

F1 **N4** R8 F8 F1
F1 **N4 N4 N4 N4**
N1 D3 C4 **N4 N4**
N1 **F1 N4** N4 R1

F1 **N4** R8 F8 F1
F1 **N4 N4 N4 N4**
N1 D3 C4 **N4 N4**
N1 **F1 N4 N4** R1

// R cell takes the maximum of the values in its row

F1 N4 R8 F8 F1
F1 N4 N4 N4 N4
N1 D3 C4 N4 N4
N1 F1 N4 N4 R4