

CmpE 150 Introduction to Computing
Fall 2015
Project 3—Due: 22/12/2015, 23:00

This is an individual project. Please do not get help from your friends. Do not use Arrays class or Recursion for this project.

Scenario: Consider a board of colors that covers an area of 16 by 16 rectangle of cells. Two cells are connected if they are neighbors either in south-north or in east-west direction. Initially each cell is either colored White (W) or Green (G).

There are four Dark Colors (DC): G, B, P, and C. And there are four Light Colors (LC): W, O, Y, and L. Everything outside the board is an O.

There are certain coloring rules for changing the type of a cell; i.e., relabelling. A cell may become:

- Black (B): If a DC is connected to 4 other DC.
- Purple (P): If a DC is connected to 3 LC, or to 2 LC and at least 1 P, or to 1 LC and at least 2 P.
- Chocolate (C): If a DC is not B and not P.
- Orange (O): If a W is connected to at least one O.
- Yellow (Y): If an O is connected to at least 2 Y and at most one O, or to 1 Y and at least 2 DC, or to at least 2 DC and at least one O.
- Light Blue (L): If a W remains unchanged till no other relabeling is possible any more.

It may happen, that after a certain cell has been relabeled, it can be relabeled once again later, because the colors of some neighbors have changed in the meantime. The colors of a board is finalized if no relabeling is possible any more or that all the cells have been processed at least three times.

Tasks: Implement a program which does the following in that order:

- Read an initial board of colors from an input file (called input.txt) and display it on the screen, together with the initial cell type statistics, as shown below.
- Explore the board and relabel the cells correctly with B, P, C, O, Y, or L according to the coloring rules until the colors are finalized.
- Display the finalized board on the screen, with the final cell type statistics, as shown below.

The screen display of the initial board with initial statistics:

```
WWWWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWWWW
GGGGGGWWWWWWWWWWWW
GGWWGGWWWWWWWWWWWW
GGWGGGWWWWWWWWWWWW
GGWWGGWWWGGGWWWW
WGGGGGGGGGGGGGGWW
WWWWWWWGGGWWWGGW
WWWWWWWGGGWWWGGW
WWWWWWWGGGWWWGGW
WWWWWWWWGGWWWGGW
WWWWWWWWWGWWWWWW
WWWWWWWWWGWWWWWW
WWWWWWWWGGGWWWW
WWWWWWWWWWWWWWWW
WWWWWWWWWWWWWWWW
INITIAL: G=61 W=193 ALL=256
```

The screen display of the explored board with statistics:

```
OOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOO
CCCCC00000000000
CCLLCC00000000000
CBPLCC00000000000
CCLLCCYYYCCCYPOO
YCCCCCCCCBCCCYO
OOOOOYCBYYYCCO
OOOOOOCBCYOCCO
OOOOOOCBBPOCCO
OOOOOYCCYOCCO
OOOOOYPYOCCO
OOOOOYPYOCCO
OOOOOOPPOCCO
OOOOOOOOOOOOOOO
OOOOOOOOOOOOOOO
EXPLORED: P=8 C=47 B=6 O=173 Y=17 L=5 ALL=256
```

Submission: You will submit a project report and your code over Moodle. Your program should be named with your initials and your student number together (e.g., OS2013800027).

Project report should consist of five sections. These are:

1. Problem Description: In this section, you should describe the problem in your words.

2. **Problem Solution:** In this section, you should specify the concepts (methods, for loop, etc.) that you use in your program. Explain each one (i.e. why you need it, what you accomplish by using it, so on.). Report how many for loops you use.
3. **Implementation:** This section will include your whole code with comments. You need to pay attention to indentation in order to improve readability.
 - Do not forget to explain each variable that you use (i.e. `int count=0; // count is the number of items`).
 - Before each method, specify what the method does (i.e. `/* This method . */`)
4. **Output of the program:** A screen-shot of your program output should be put in this section.
5. **Conclusion:** You should evaluate your work here. State whether you have solved the problem correctly. If not, state what is missing, what could have been improved, and so on.

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.

Partial Submission: If you cannot do everything above, you should still submit your code as well as your report. Try to do as much as possible.

Late Submission: Any submission after the deadline is considered late and will not be accepted.