REPORT

1.PROBLEM DESCRIPTION

In this problem ,there is a board of colors that covers an area of 16 by 16 rectangle of cells. First, it is necessary to read the initial board whose each cell is either colored White(W) or Green(G) from an input file(input.txt) and display it on the screen with the initial cell type statistics. 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. And then,it is asked to change the colors of each cells with the following rules:

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.

Finally, the finalized board must be displayed on the screen,with the final cell type statistics.

2.PROBLEM SOLUTION

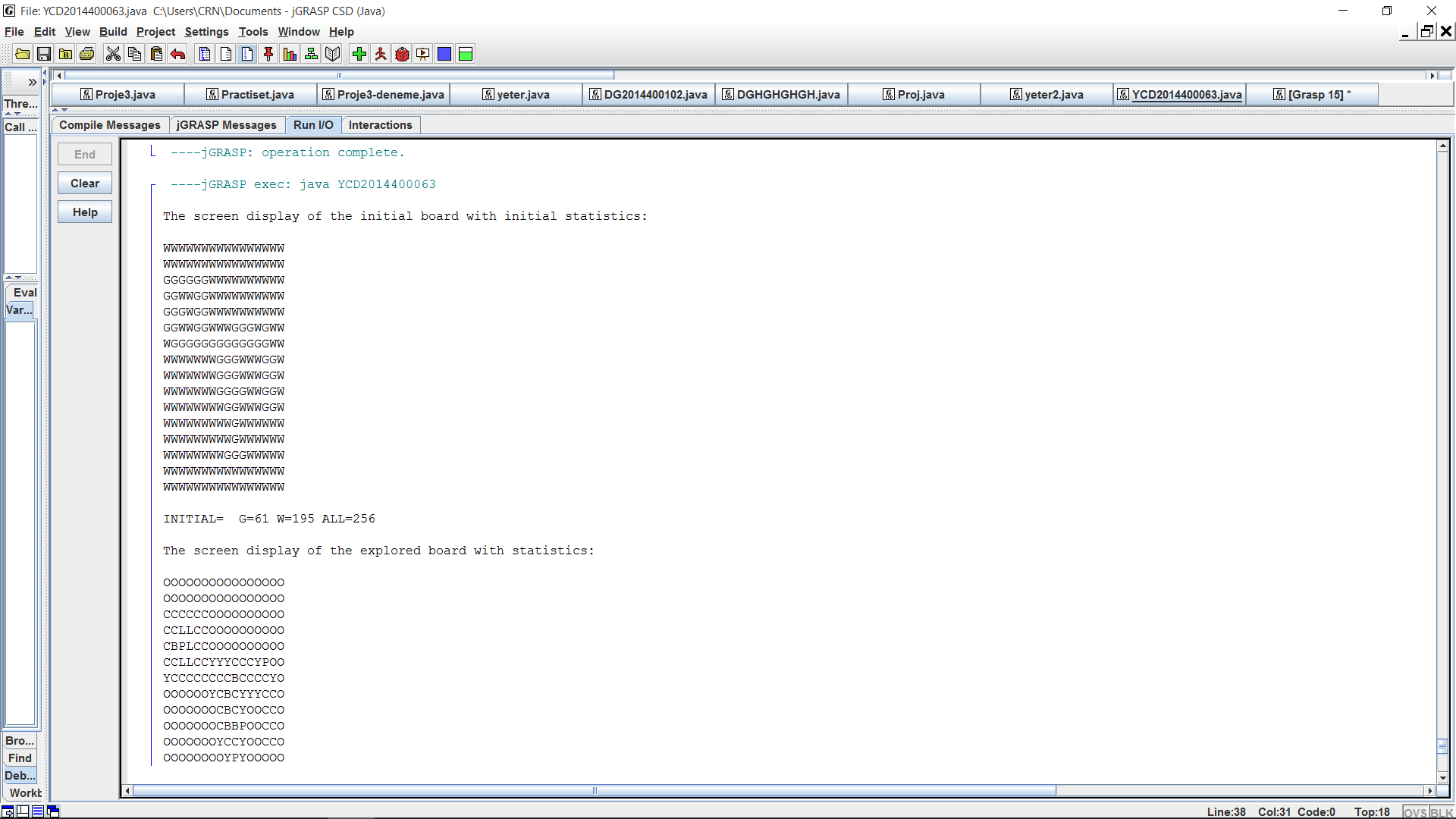
In the main method, I used one Scanner statement to read the board from the file. I also used 2 String arrays,one for the read board from the file and it’s size 3\*18, because I take the rows as one token and also the board must be centered when I am seperating the rows (it contains 16 colors but in one string)to the 16 colors. Second string array is to store the board as a way that the each color can be changed according to the rules.I used to nested for loops ,first is to read file and store the colors as a rows and the second is to seperate them. In order to make changes, I used while loop. While loop provides to make changes until the statistics equal to the statistics of previous board. There is an nested loop inside the while,it is to check each cell on the board and if the situation of the cells that is checking and the situation of it’s neighbors fits with the rules for relabeling,it changes the color of the cell. Inside the second loop i assigned the cell to check\_b, and its neighbors to the top,left,right and bottom. And also, I assigned 20 booleans to check the situations which are used in the if/else statements. To check rules, first i checked the color of cells is one of the dark colors. If this test is correct, the other if/else/if statements check the possibility of black,purple and chocolate one by one. If the color of cell is not one of the dark colors, it goes into else statement . In the else statement,it checks the possibility of orange and yellow in order. When there is no other relabeling is possible any more,it goes on with another nested for loop to change all unchanged “W” s to the “L”. At the end of the main method, I print the board by calling printBoard method and print the color statistics by calling the counter method.

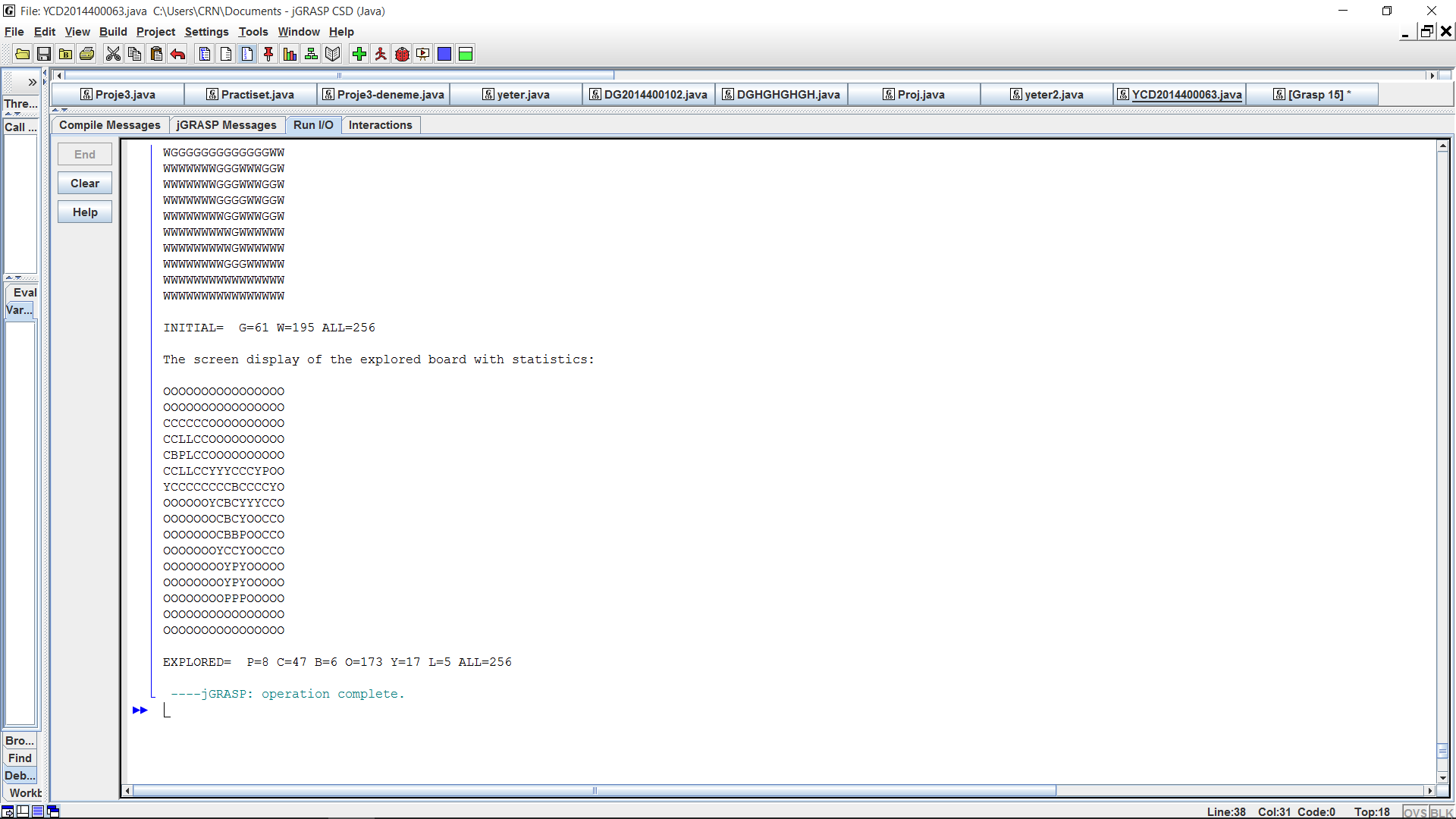
I used two methods. One is to print board by sending the board arrayand the size of the array. I used nested for loop and it prints each cell one by one.Second method is to determine the board’s color statistics and to print these statistics. I used one integer string to count the number of the each color that board contains. I also used a string array that fits with the integer array. For example the first cell in the integer array represents the number of the color that is situated in the first cell in the string array. I used a nested loop the determine the number colors and i stored in the integer array. Then i print the statistics by using while loop. While loop will work when the size of an integer a is smaller than the length of the integer array. And if the cell that is checking is not zero,which means the board contains that color, it will print the color type by looking the string array and it prints the number of each color.

3.IMPLEMENTATION

1 import java.util.\*; // This statement is to get cells from the file.  
 2 import java.io.\*;//This statement is to use file.  
 3   
 4 public class YCD2014400063 {  
 5   
 6 public static void main(String [] args)throws FileNotFoundException {//The main method provides to get the board from the file,to arrange the String array for board cells  
 7 //and to print board and statistics by calling printBoard and counter methods and to make changes to the board according to the rules  
 8 //throws FileNotFoundException provides to compile class without giving an file not found error  
 9 Scanner console=new Scanner(new File("input.txt")); //This scanner is to read file by giving current directory   
 10 int number1=16; //number1 =to determine the size of the board  
 11 int number2=1; //number2= to determine the columns that rows are assigned initially because it takes the rows as a one string at the beginning  
 12 String[][] board1 = new String[number1+2][number2+2]; //board1 is the String array whose size 3\*18,is to get rows from file initially as a one string and it has to be centered  
 13 String[][] board2 = new String[number1+2][number1+2]; //board2 is the String array whose size is 18\*18is to seperate each color from one string and replace "O" outside the board  
 14   
 15 for(int i=1;i<number1+1;i++){ //this nested for loop reads file initially   
 16   
 17 for(int j=1;j<number2+1;j++){  
 18   
 19 board1[i][j]= console.next();//it takes 16 colors as a one string  
 20   
 21 }  
 22 }   
 23 String row=""; //row is a empty string that takes the next row as a one string  
 24 for(int i=0;i<18;i++){ // this nested for loop is to seperate one string to the 16 colors  
 25   
 26 for(int j=0;j<18;j++){  
 27 row=board1[i][1]+" "; //" " is to prevent exception  
 28 if(i==0 ||i==17|| j==0|| j==17){  
 29 board2[i][j]="O"; // to place "O" around the board  
 30 }  
 31 else if(j>0 && j<17){  
 32 board2[i][j]=row.substring(j-1,j);//if cell coordinates do not equal to the edge coordinates ,cell is equal to the row's one color  
 33 }  
 34 }  
 35   
 36 }  
 37 System.out.println("The screen display of the initial board with initial statistics:");  
 38 System.out.println();  
 39 printBoard(board2,number1);//to call printBoard method for printing initial board   
 40 System.out.print("INITIAL= ");   
 41 System.out.println(counter(board2,number1)); //to call counter method to print color statistics  
 42 System.out.println();   
 43 String check1=counter(board2,number1);  
 44 String check2="";  
 45   
 46   
 47   
 48 while(!check2.equals(check1)){ //while provides to exit from loop if there no changes to make  
 49   
 50 for(int i=1;i<17;i++){ //this nested loop provides to check each cell whether it is going to change  
 51   
 52 for(int j=1;j<17;j++){  
 53   
 54 String check\_b=board2[i][j]; //check\_b=cell that is going to be checked  
 55 String top=board2[i-1][j]; //the cell's north neighbor  
 56 String left=board2[i][j-1]; //the cell's west neighbor  
 57 String right=board2[i][j+1]; //the cell's east neighbor  
 58 String bottom=board2[i+1][j];//the cell's south neighbor  
 59 boolean c1 =top.equals("G")|| top.equals("B")|| top.equals("P")|| top.equals("C"); //to check north neighbor is one of the dark colors  
 60 boolean c2= left.equals("G")||left.equals("B")||left.equals("P")||left.equals("C");//to check west neighbor is one of the dark colors  
 61 boolean c3=right.equals("G")||right.equals("B")||right.equals("P")||right.equals("C");//to check east neighbor is one of the dark colors  
 62 boolean c4= bottom.equals("G")||bottom.equals("B")||bottom.equals("P")||bottom.equals("C");//to check south neighbor is one of the dark colors  
 63 boolean p1=top.equals("W")|| top.equals("O")|| top.equals("Y")|| top.equals("L");//to check north neighbor is one of the light colors  
 64 boolean p2=left.equals("W")||left.equals("O")||left.equals("Y")||left.equals("L");//to check west neighbor is one of the light colors  
 65 boolean p3=right.equals("W")||right.equals("O")||right.equals("Y")||right.equals("L");//to check east neighbor is one of the light colors  
 66 boolean p4=bottom.equals("W")||bottom.equals("O")||bottom.equals("Y")||bottom.equals("L");//to check south neighbor is one of the light colors  
 67 boolean p5=top.equals("P");//to check north neighbor equals to "P"  
 68 boolean p6=right.equals("P");//to check north neighbor equals to "P"  
 69 boolean p7=left.equals("P");//to check north neighbor equals to "P"  
 70 boolean p8=bottom.equals("P");//to check north neighbor equals to "P"  
 71 boolean y1=top.equals("Y");//to check north neighbor equals to "Y"  
 72 boolean y2=left.equals("Y");//to check west neighbor equals to "Y"  
 73 boolean y3=right.equals("Y");//to check east neighbor equals to "Y"  
 74 boolean y4=bottom.equals("Y");//to check south neighbor equals to "Y"  
 75 boolean y9=top.equals("O");//to check north neighbor equals to "O"  
 76 boolean y10=left.equals("O");//to check west neighbor equals to "O"  
 77 boolean y11=right.equals("O");//to check east neighbor equals to "O"  
 78 boolean y12=bottom.equals("O");//to check south neighbor equals to "O"  
 79   
 80   
 81 if(check\_b.equals("G")||check\_b.equals("B")||check\_b.equals("P")||check\_b.equals("C")){//if statement to check cell is one of the dark colors  
 82 //checking boaard for black  
 83 if((c1&&c2&&c3&&c4)){ //if cell is connected to 4 other dark colors,it will change cell to the "B"  
 84   
 85 board2[i][j]="B";  
 86   
 87 }  
 88 //checking board for purple  
 89 else if((p1&&p2&&p3&&!p4)||(p1&&!p2&&p3&&p4)||(p1&&p2&&!p3&&p4)||(!p1&&p2&&p3&&p4)){//if the cell is connected to 3 light colors,it will change cell to the "P"  
 90 board2[i][j]="P";  
 91 }//if the cell is connected to 2 LC and at least 1 P,it will change cell to the "P"  
 92 else if(((p1&&p2&&!p3&&!p4)||(p1&&!p2&&p3&&!p4)||(p1&&!p2&&!p3&&p4)||(!p1&&p2&&p3&&!p4)||(!p1&&p2&&!p3&&p4)||(!p1&&!p2&&p3&&p4))&&(p5||p6||p7||p8)){  
 93   
 94 board2[i][j]="P";  
 95 }//if the cell is connected to 1 LC and at least 2 P,,it will change cell to the "P"  
 96 else if(((p1&&!p2&&!p3&&!p4)||(!p1&&p2&&!p3&&!p4)||(!p1&&!p2&&p3&&!p4)||(!p1&&!p2&&!p3&&p4))&&(!(!p5&&!p6&&!p7&&!p8)&&!((p5&&!p6&&!p7&&!p8)||(!p5&&p6&&!p7&&!p8)||(!p5&&!p6&&p7&&!p8)||(!p5&&!p6&&!p7&&p8)))){  
 97   
 98 board2[i][j]="P";  
 99   
100 }  
101 //checking for chocolate  
102 else if(!check\_b.equals("B")&&!check\_b.equals("P")){ //If the cell is not B and not P,,it will change cell to the "C"   
103   
104 board2[i][j]="C";  
105 }  
106   
107 }  
108 //checking for orange  
109 else {//if the cell does not equal to the one of the dark colors   
110   
111 if(check\_b.equals("W")&&(!(!y9&&!y10&&!y11&&!y12))){//If a W is connected to at least one O,,it will change cell to the "O"  
112   
113 board2[i][j]="O";  
114   
115 }   
116 //checking for yellow  
117 else if(check\_b.equals("O")){//If the cell equals to the "O"  
118 //If the cell is connected to at least 2 Y and at most one O,it will change cell to the "Y"  
119 if((!(!y1&&!y2&&!y3&&!y4)&&!((y1&&!y2&&!y3&&!y4)||(!y1&&y2&&!y3&&!y4)||(!y1&&!y2&&y3&&!y4)||(!y1&&!y2&&!y3&&y4)))&&((!y9&&!y10&&!y11&&!y12)||((y9&&!y10&&!y11&&!y12)||(!y9&&y10&&!y11&&!y12)||(!y9&&!y10&&y11&&!y12)||(!y9&&!y10&&!y11&&y12)))){  
120   
121 board2[i][j]="Y";   
122 }  
123 //If the cell is connected to 1 Y and at least 2 DC,it will change cell to the "Y"  
124 else if(((y1&&!y2&&!y3&&!y4)||(!y1&&y2&&!y3&&!y4)||(!y1&&!y2&&y3&&!y4)||(!y1&&!y2&&!y3&&y4))&&(!(!c1&&!c2&&!c3&&!c4)&&!((c1&&!c2&&!c3&&!c4)||(!c1&&c2&&!c3&&!c4)||(!c1&&!c2&&c3&&!c4)||(!c1&&!c2&&!c3&&c4)))){  
125   
126 board2[i][j]="Y";  
127   
128 }  
129 //If the cell is connected to at least 2 DC and at least one O,it will change cell to the "Y"  
130 else if(!(!c1&&!c2&&!c3&&!c4)&&!((c1&&!c2&&!c3&&!c4)||(!c1&&c2&&!c3&&!c4)||(!c1&&!c2&&c3&&!c4)||(!c1&&!c2&&!c3&&c4))&&(!(!y9&&!y10&&!y11&&!y12))){  
131   
132 board2[i][j]="Y";  
133   
134 }  
135 }   
136 }   
137 }   
138 }  
139   
140 check2=check1;  
141 check1=counter(board2,number1);  
142   
143 }   
144   
145 for(int i=1;i<number1+1;i++){//this for loop is to change W ,that remains unchanged till no other relabeling is possible any more,to the "L".  
146   
147 for(int j=1;j<number1+1;j++){  
148   
149 if(board2[i][j].equals("W")){  
150   
151 board2[i][j]="L";  
152   
153 }  
154 }  
155 }  
156 System.out.println("The screen display of the explored board with statistics:");  
157 System.out.println();  
158 printBoard(board2,number1); //to print final boar by calling printBoard method   
159 System.out.print("EXPLORED= ");   
160 System.out.println(counter(board2,number1));//to print final color statistics by calling counter method  
161   
162 }  
163 //This method is to count how many colors the board contains ,to print the color statistics  
164 public static String counter(String[][] board,int n){  
165   
166 int[] count= new int[8]; //this array provides to count how many color token board consists of.   
167 String[] countletter={"G","W","P","C","B","O","Y","L"};// element 0 is to count G,element 1 is to count B,element 2 is to count P,element 3 is to count C,   
168 //element 4 is to count W,element 5 is to count O,element 6 is to count Y,element 7 is to count L,  
169 //string array is to match values from the integer array and the colors  
170 for(int i=1;i<n+1;i++){ //the nested for loop to count how many colors the board contains  
171 for(int j=1;j<n+1;j++){   
172 for(int k=0;k<8;k++){ //this for loop is to find each cell's color  
173 if(board[i][j].equalsIgnoreCase(countletter[k])){//to check cell's color is equal to the color that the string array countLetter contains   
174   
175 count[k]++; //to increases the color number which matches with the color name from the countLetter array  
176   
177 }   
178 }   
179 }  
180 }  
181 int a=0; //to provide test for while loop  
182 String statistics="";  
183 while(a<count.length){ //to print all the colors that the board contains  
184   
185 if(count[a]!=0){ //if the board contains a that color,the cell that is assigned for that color will not equal to the zero  
186   
187 statistics += countletter[a]+"="+count[a]+" ";//to print the color and how many cells include this color  
188 }   
189 a++; //to check next color  
190 }  
191 return statistics + "ALL="+ n\*n;//to print the size of the board  
192 }  
193   
194 public static void printBoard(String[][]board,int number1){//to print board  
195   
196 for(int i=1;i<number1+1;i++){ //this nested for loop is to print board,i=line  
197   
198 for(int j=1;j<number1+1;j++){//j=column  
199   
200 System.out.print(board[i][j]);//it prints the each cell from one row  
201   
202 }  
203 System.out.println();   
204 }   
205 System.out.println();  
206 }   
207 }

4.OUTPUT OF THE PROGRAM





5.CONCLUSION

I solved the problem.