```
public class ColorGrid
{
    private String[][] myPixels;
    private int myRows;
    private int myCols;

    /**
     * Creates numRows × numCols ColorGrid from String s.
     * @param s the string containing colors of the ColorGrid
     * @param numRows the number of rows in the ColorGrid
     * @param numCols the number of columns in the ColorGrid
     */
    public ColorGrid(String s, int numRows, int numCols)
    { /* to be implemented in part (a) */ }

    /**
     * Precondition:  myPixels[row][col] is oldColor, one of "r",
     *                "b","g", or "y".
     *                newColor is one of "r","b","g", or "y".
     * Postcondition: if 0 <= row < myRows and 0 <= col < myCols,
     *                paints the connected region of
     *                myPixels[row][col] the newColor.
     *                Does nothing if oldColor is the same as
     *                newColor.
     * @param row the given row
     * @param col the given column
     * @param newColor the new color for painting
     * @param oldColor the current color of myPixels[row][col]
     */
    public void paintRegion(int row, int col, String newColor,
        String oldColor)
    { /* to be implemented in part (b) */ }

    //other methods not shown
        ...
}
```

(a) Write the implementation code for the `ColorGrid` constructor. The constructor should initialize the `myPixels` matrix of the `ColorGrid` as follows: The dimensions of `myPixels` are `numRows` × `numCols`. `String s` contains `numRows` × `numCols` characters, where each character is one of the colors of the grid—`"r"`, `"g"`, `"b"`, or `"y"`. The characters are contained in `s` row by row from top to bottom and left to right. For example, given that `numRows` is 3, and `numCols` is 4, if `s` is `"brrygrggyyyr"`, `myPixels` should be initialized to be

```
b r r y
g r g g
y y y r
```