Eleanor Radaza
201810769

1. Cite portions of your solution (via screenshots and explanations) wherein you were able to **reuse code**; must have at least one instance *(affects quality metric)*

Clearing the Google spreadsheet, that is, removing all other sheets except for “Main Sheet” and “Results” is done quite frequently throughout the solution. Below is the helper function that is called whenever the extra sheets need to be deleted, specifically, when the “Remove Extra Sheets” functionality is selected and whenever we start the “Group by Verdict” procedure.

```
function deleteExtraSheets() {
    sheets = SpreadsheetApp.getActiveSpreadsheet().getSheets();
    for (i = 0; i < sheets.length ; i++ ) {
      sheet = sheets[i];

      if (sheet.getSheetName() != "Main Sheet" && sheet.getSheetName()  != "Results") {
        SpreadsheetApp.getActiveSpreadsheet().deleteSheet(sheet);
      }
    }
}
```

Below is a code from https://stackoverflow.com/questions/33787057/how-to-set-a-sheets-number-of-rows-and-columns-at-creation-time-and-a-word-ab that is also heavily used in the “Group by Verdict” procedure

```
/**
 * Wrapper for Spreadsheet.insertSheet() method to support customization.
 * All parameters are optional & positional.
 *
 * @param {String}  sheetName     Name of new sheet (defaults to "Sheet #")
 * @param {Number}  sheetIndex    Position for new sheet (default 0 means "end")
 * @param {Number}  rows          Vertical dimension of new sheet (default 0 means "s
ystem default", 1000)
 * @param {Number}  columns       Horizontal dimension of new sheet (default 0 means
"system default", 26)
 * @param {String}  template      Name of existing sheet to copy (default "" means no
ne)
 *
 * @returns {Sheet}               Sheet object for chaining.
 */
function insertSheet( sheetName, sheetIndex, rows, columns, template ) {

  // Check parameters, set defaults
  ss = SpreadsheetApp.getActive();
  numSheets = ss.getSheets().length;
  sheetIndex = sheetIndex || (numSheets + 1);
  sheetName = sheetName || "Sheet " + sheetIndex;
  options = template ? { 'template' : ss.getSheetByName(template) } : {};
```

Eleanor Radaza
201810769

```
  // Will throw an exception if sheetName already exists
  newSheet = ss.insertSheet(sheetName, sheetIndex, options);

  if (rows !== 0) {
    // Adjust dimension: rows
    newSheetRows = newSheet.getMaxRows();

    if (rows < newSheetRows) {
      // trim rows
      newSheet.deleteRows(rows+1, newSheetRows-rows);
    }
    else if (rows > newSheetRows) {
      // add rows
      newSheet.insertRowsAfter(newSheetRows, rows-newSheetRows);
    }
  }

  if (columns !== 0) {
    // Adjust dimension: columns
    newSheetColumns = newSheet.getMaxColumns();

    if (columns < newSheetColumns) {
      // trim rows
      newSheet.deleteColumns(columns+1, newSheetColumns-columns);
    }
    else if (columns > newSheetColumns) {
      // add rows
      newSheet.insertColumnsAfter(newSheetColumns,columns-newSheetColumns);
    }
  }

  // Return new Sheet object
  return newSheet;
}
```

Eleanor Radaza
201810769

Since clearing the results page is done whenever we are fetching new data from OJ and when "Clear Results" functionality is selected, the helper function below is made for easy reference.

```
function clearResultsNoPrompt(){
  resultsSheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("Results");
  resultsSheet.activate();

  numRows = resultsSheet.getLastRow();
  Logger.log("Number of Rows:" + numRows);
  numCols = resultsSheet.getLastColumn();
  Logger.log("Number of Cols:" + numCols);

  if (numRows >= 2 && resultsSheet.getRange("A2:F2").getDisplayValues() != [['Date','Problem','User','Language','Results',
'Points'],['','','','','','']]) {
    resultsSheetDataRange = resultsSheet.getRange(3,1,numRows - 2, numCols);
    resultsSheet.deleteRows(2,numRows - 2);

    resultsSheetDataRange = resultsSheet.getRange("A2:F2");
    resultsSheetDataRange.setValues([['','','','','','']]);
  }
}
```

2. **(Bonus: 5 pts)** How did you find the exercise in terms of enjoyment, ease, learning, usefulness, etc.? How could this exercise be improved (optional)?

   I enjoyed the exercise because it was challenging.

   Coding exercises are always fun when I have the least bit of idea what to do. Even when I had to reorient myself JavaScript, and at the same time learn Google Apps Script, the problem, it's solution and the language I am using are all intuitive. I feel like the exercise provided the right amount of challenge without being too harsh.

   Moreover, the exercise is still doable even with a tight schedule. That is definitely a plus. Although, I don't think that should be something I look forward to in future exercises. Still, that fact is much appreciated.

   I do appreciate the reminder to write cleaner code because that is an area I should really improve on.

   I also loved seeing Google Apps Script, in general, for the very first time. It excites me to think about the things I can do with more ease with this new knowledge in my hands.