**Lab 4b: Bubble Factory**

Please check the code below:

Note: I have written comments on why the code was typed this way and my reference was the lecture and the textbook.

For a better clean code, please check the github repository in this link:

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8" />

        <title>Bubble Factory Test Lab</title>

    </head>

    <body>

        <script>

            var scores = [

                60, 50, 60, 58, 54, 54, 58, 50, 52, 54, 48, 69, 34, 55, 51, 52, 44, 51,

                69, 64, 66, 55, 52, 61, 46, 31, 57, 52, 44, 18, 41, 53, 55, 61, 51, 44,

            ];

            function printAndGetHighScore(scores) {

                //highScore is a variable to hold the high score

                var highScore = 0;

                //the output variable is just to print out in the console each and every score in a separate line.

                var output;

                //Here as mentioned in page 137 by Judy ;), we will need to iterate through the scores array to render a list of each individual index

                //Mentioned as well in page 148, "Check each time through the loop to see if we have a higher score, and if so, that's our new high score"

                // using for loop in here as mentioned in the lecture, it is just an easier syntax than the while loop. stated in the textbook as well in page 140 & 145 by the for loop itself LOL "When we use a WHILE loop, you have to initialize your counter and increment your counter in separate statements.....".

                // it was funny reading the conversation between the While loop and For loop in page 144 & 145 though LOL

                // mentioned in the lecture, the increment could be i++ or i=i+1

                for (var i = 0; i < scores.length; i++) {

                    output = "Bubble solution #" + i + " score: " + scores[i];

                    console.log(output);

                    //Here, there is a condition to overwrite the highscore, for example the 1st iteration i=0, which by substitution, the condition will look like scores[0] > highscore which is 60 > 0, this is true, therefore, the body of the if condition will be executed so highScore will be equal 60.

                    //The 2nd iteration i=1 which by substitution scores[1] > 60 => 50 > 60 which is false, then highScore will be as is and loop will go on until it drops out when i < scores.length

                    if (scores[i] > highScore) {

                        highScore = scores[i];

                    }

                }

                return highScore;

            }

            function getBestResults(scores, highScore) {

                //Now, we will start with an empty array to hold the solutions with the highest scores, and add each solution that has that high score one at a time to it as we iterate through the scores array

                var bestSolutions = [];

                for (var i = 0; i < scores.length; i++) {

                    //Each time through the loop, we will compare the score at index i with the highScore(which was stored by using function printAndGetHighScore) and if they are equal, then we add that index to the bestSolutions array by using push

                    if (scores[i] == highScore) {

                        // Added the "#" symbol to make the output/ report looks the same as in page 135 in the textbook

                        //I like it this way better than the output in page 150

                        bestSolutions.push(" #" + i);

                    }

                }

                return bestSolutions;

            }

            //calling printAndGetHighScore will return highscore as well as there will be an output console.log("all available scores in a list") should show in the console.

            // but in the end, highScore variable will only have one value (Integer) because printAndGetHighScore function return this value only.

            var highScore = printAndGetHighScore(scores);

            //As mentioned in page 148, it is required in the report to get the overall score count which is by simply using scores.length

            console.log("Bubble tests: " + scores.length);

            // Printing highScore

            console.log("Highest bubble score: " + highScore);

            //

            var bestSolutions = getBestResults(scores, highScore);

            console.log("Solutions with the highest score:" + bestSolutions);

        </script>

    </body>

</html>

Graphical user interface, application, table

Description automatically generated