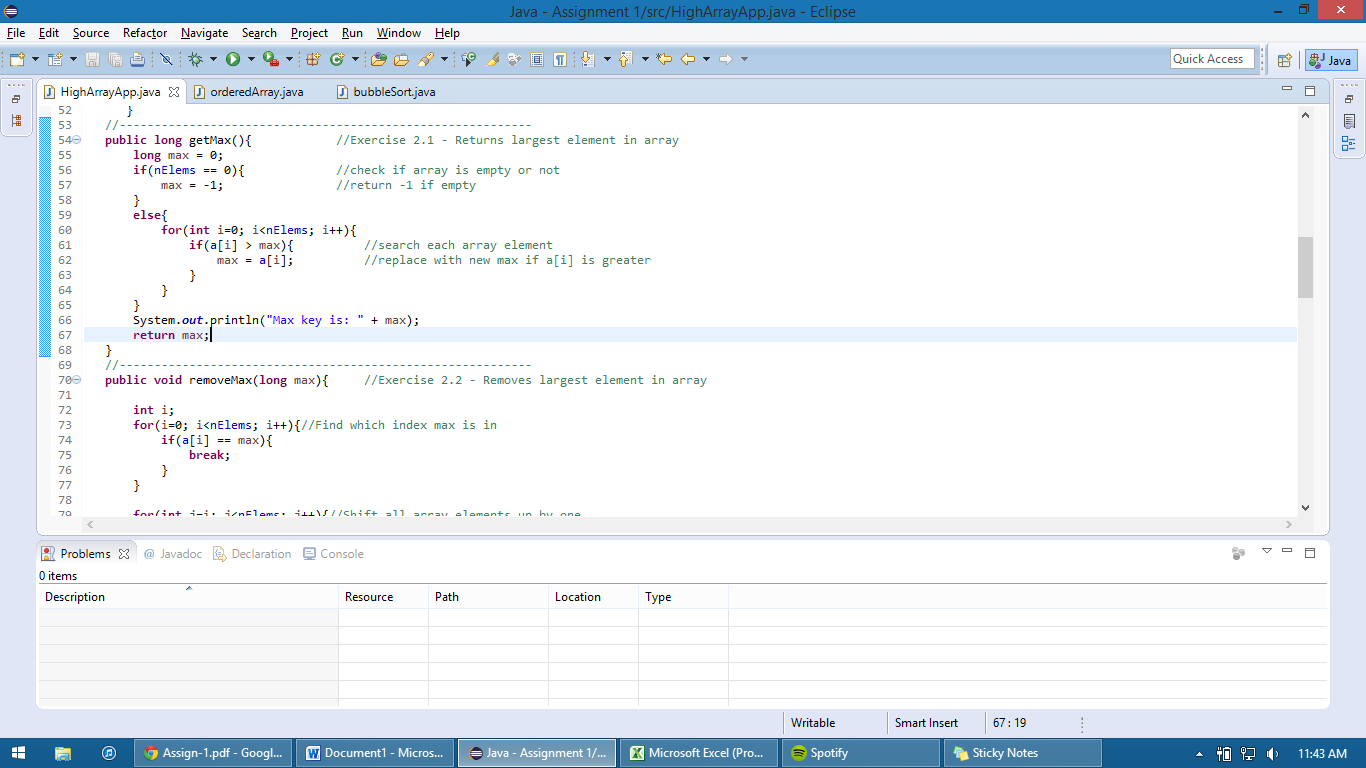
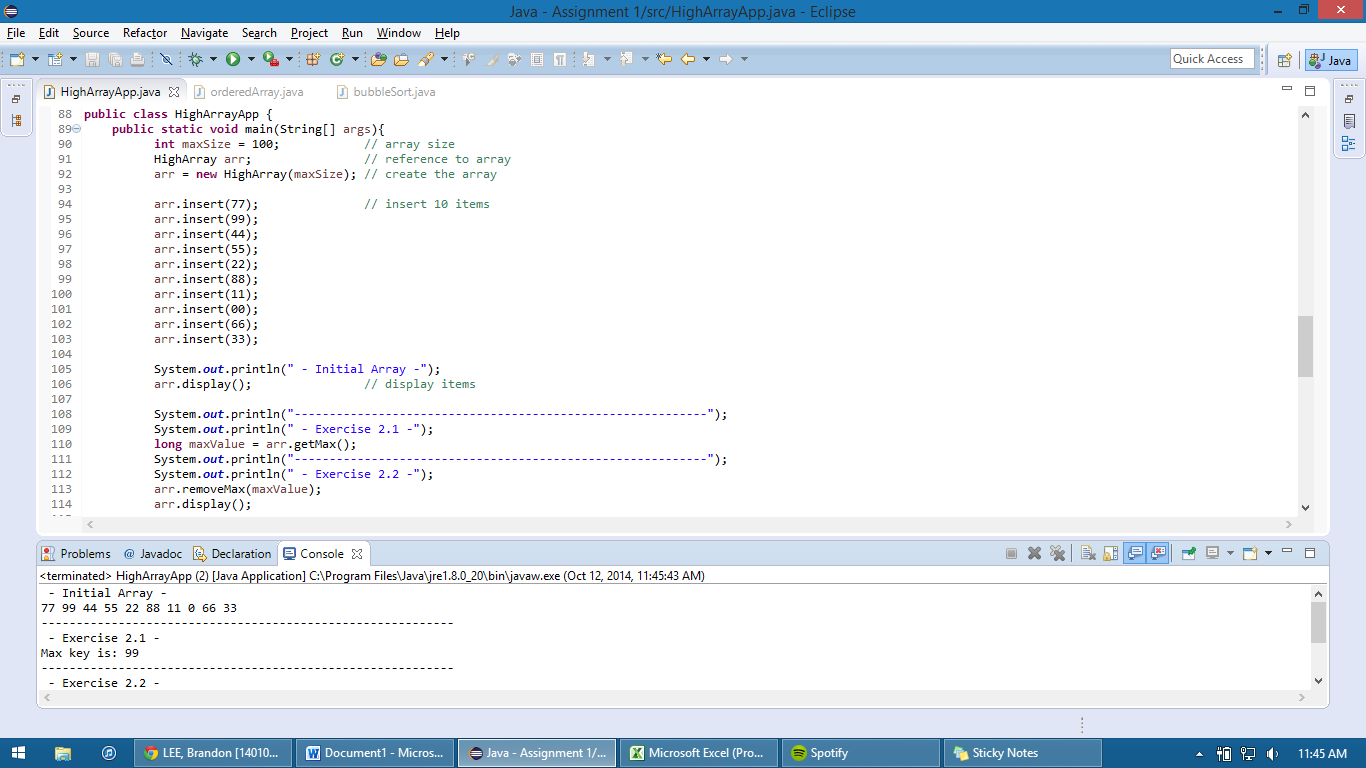
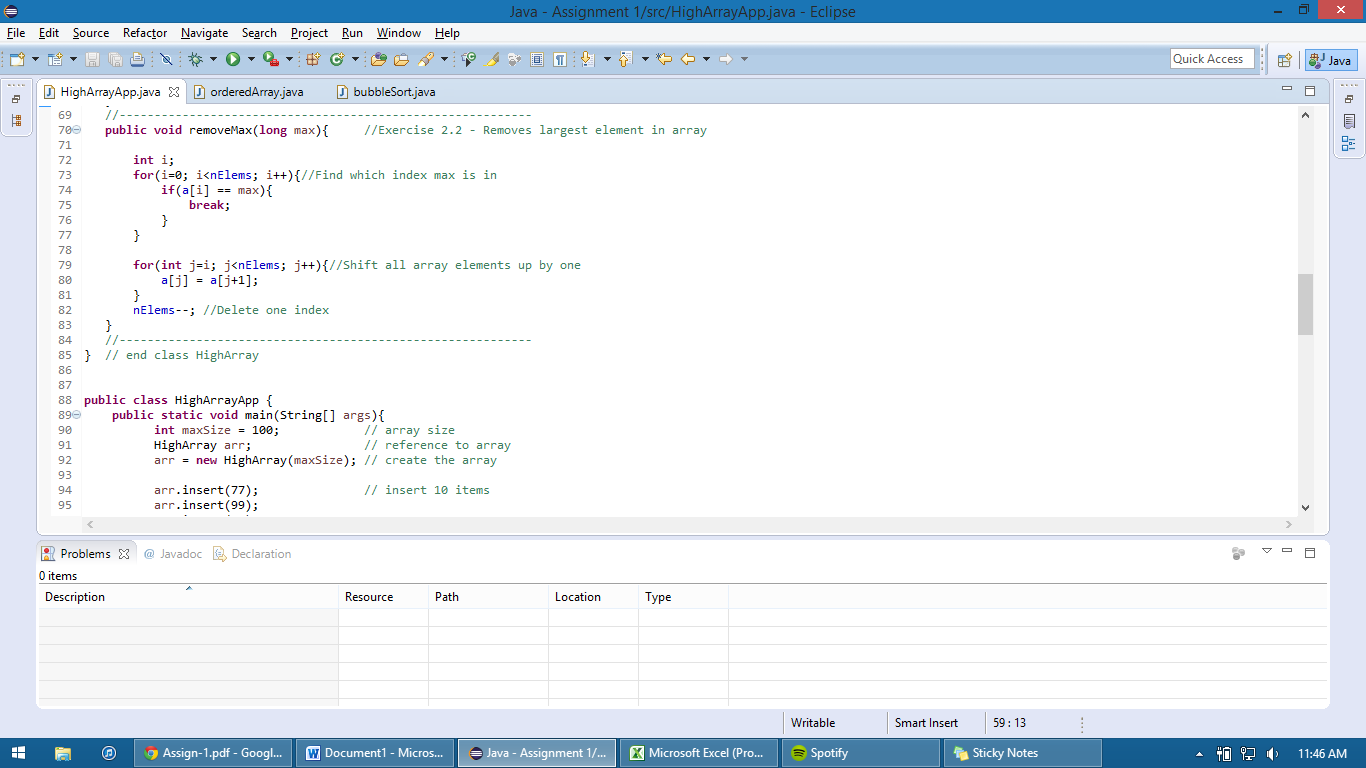
Brandon Lee  
14010627X  
Assignment 1  
10/12/14

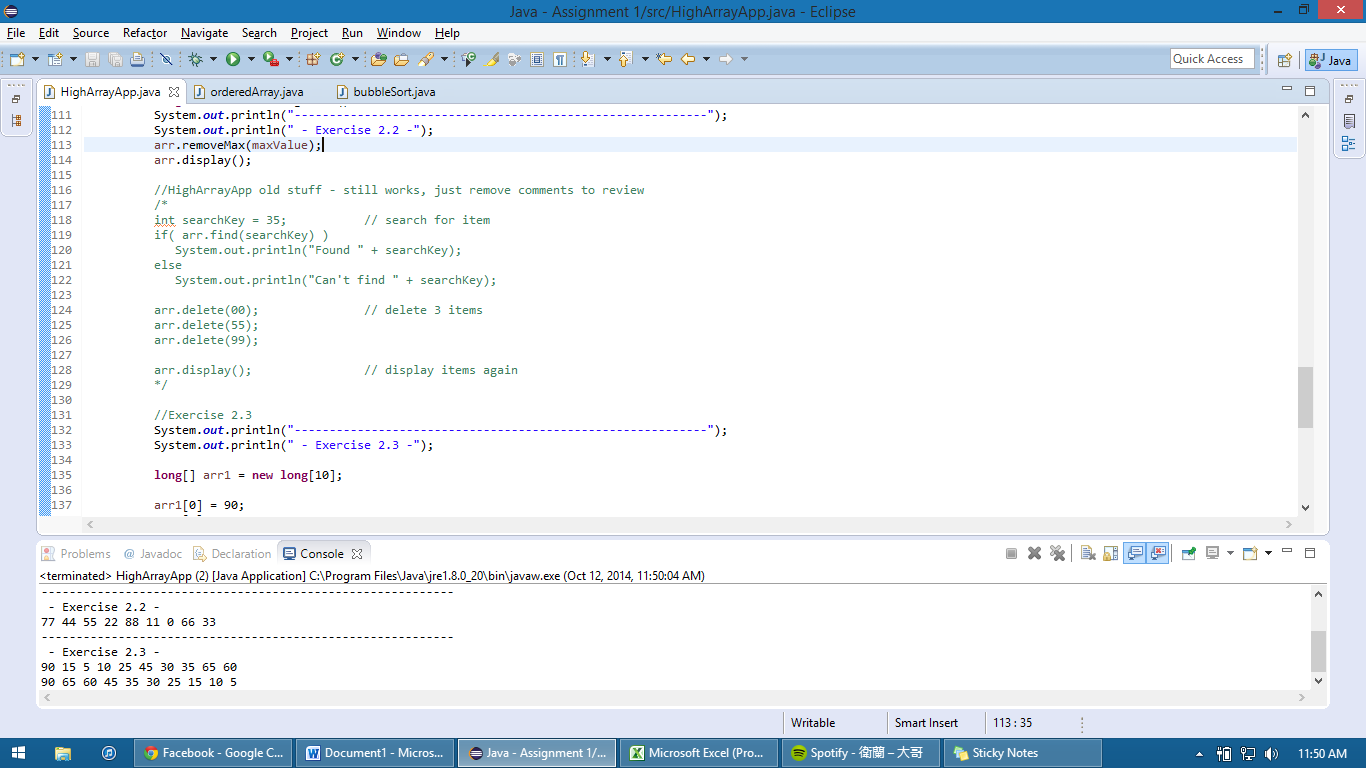
\*\*\* I wasn’t exactly sure what I need to capture with the screenshots, but I have included all the code nevertheless into my submission for reference. Table for Q5 is included at end of report. Please contact me if there is any difficulty in evaluating the code or if there is any trouble with this report.

2.1 Screenshots

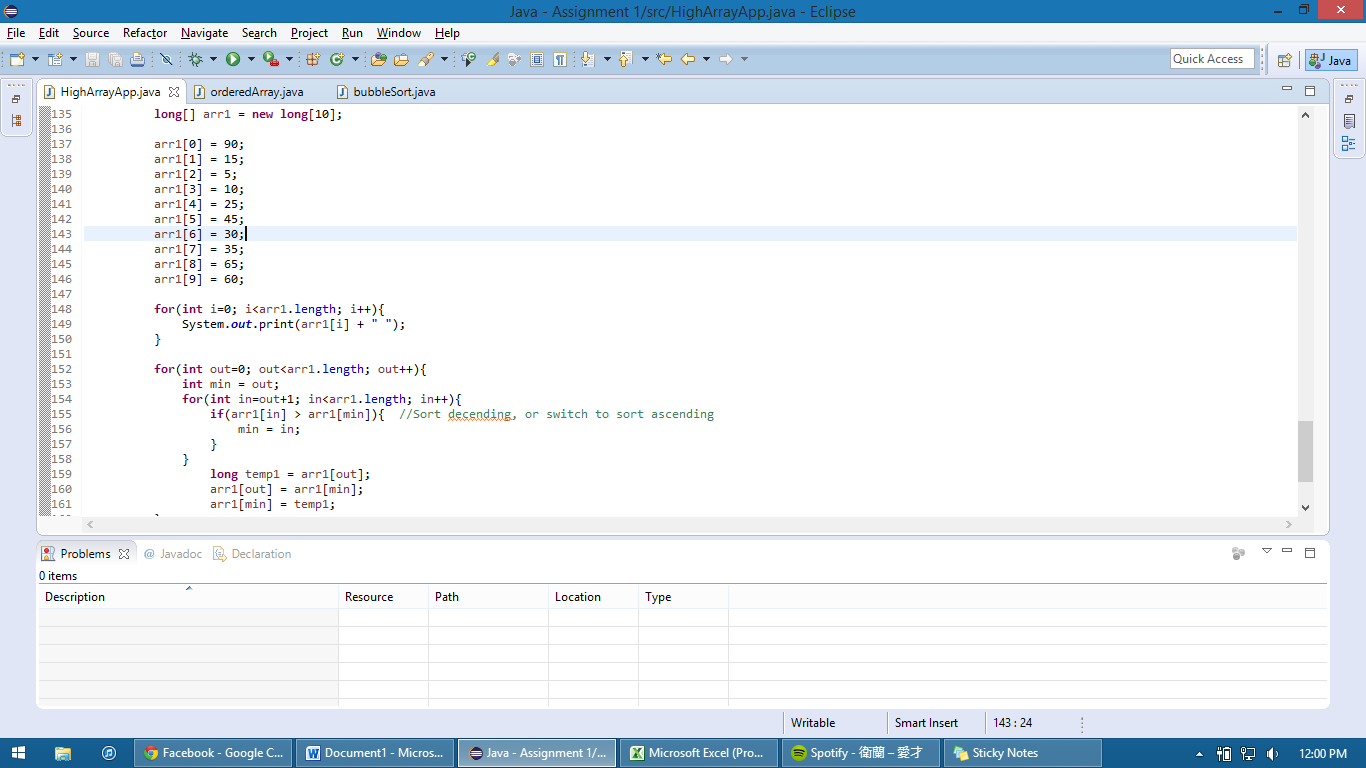


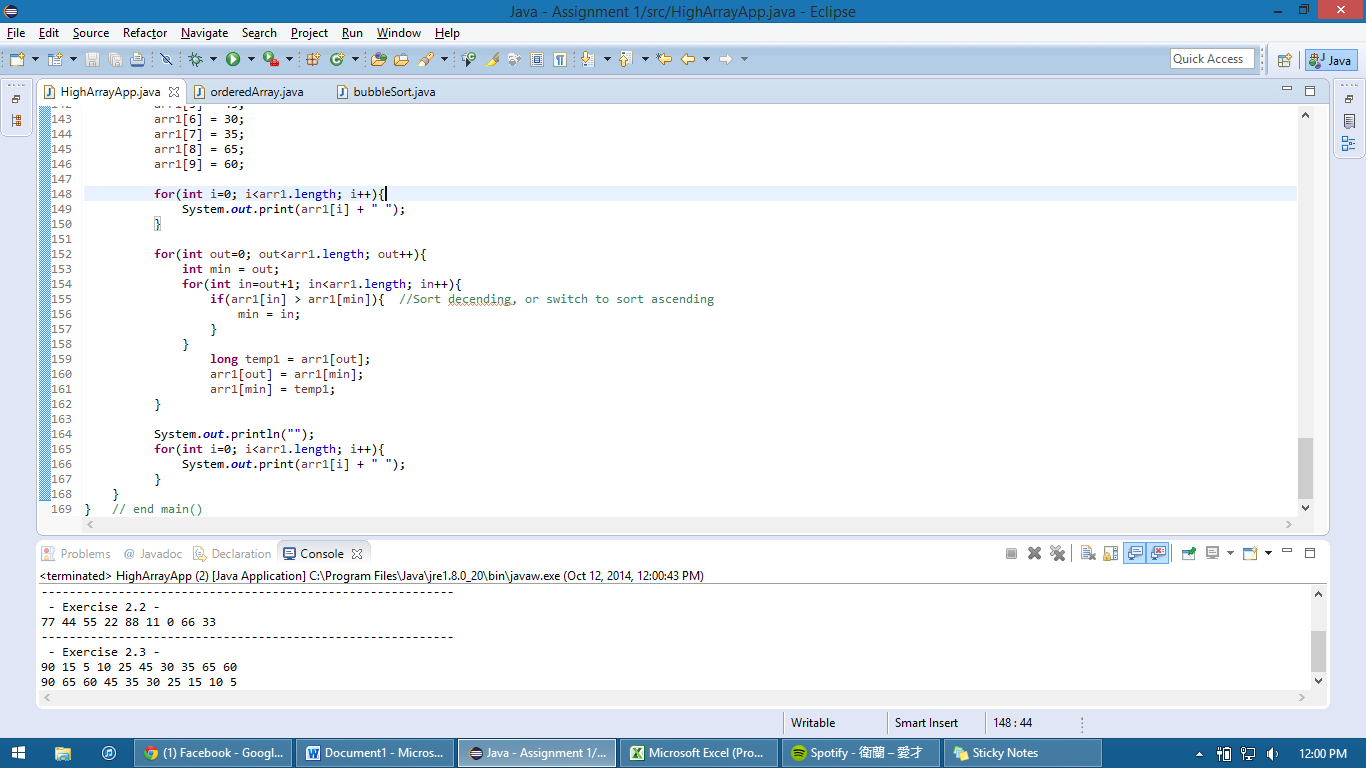


2.2 Screenshots

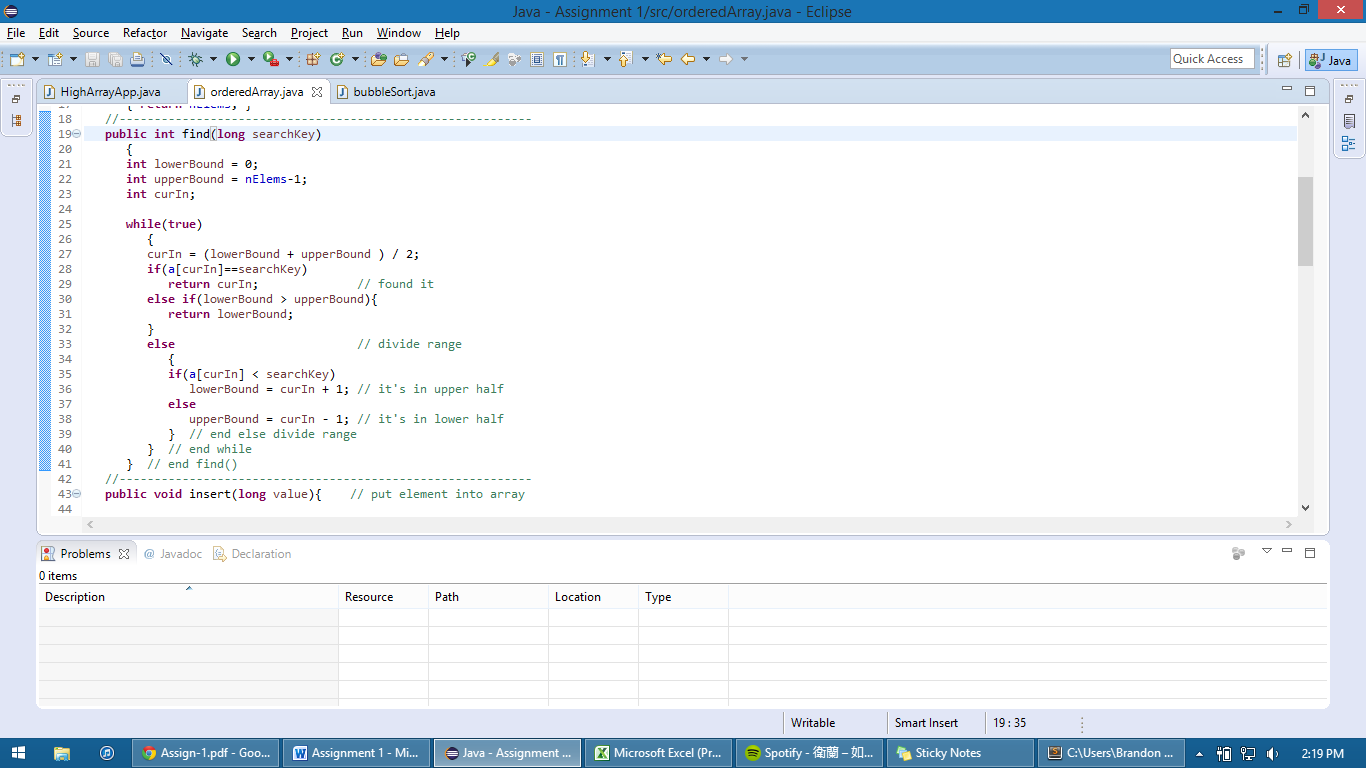


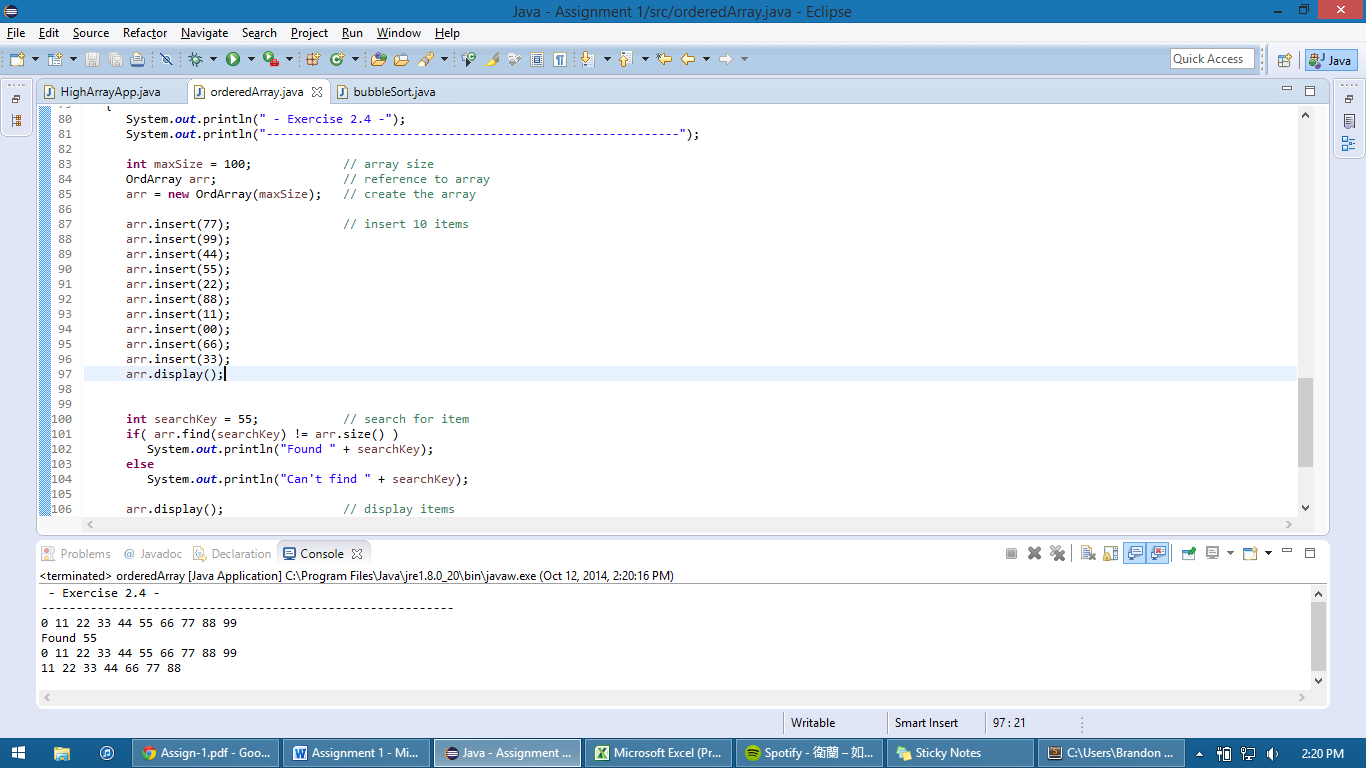
2.3 Screenshots



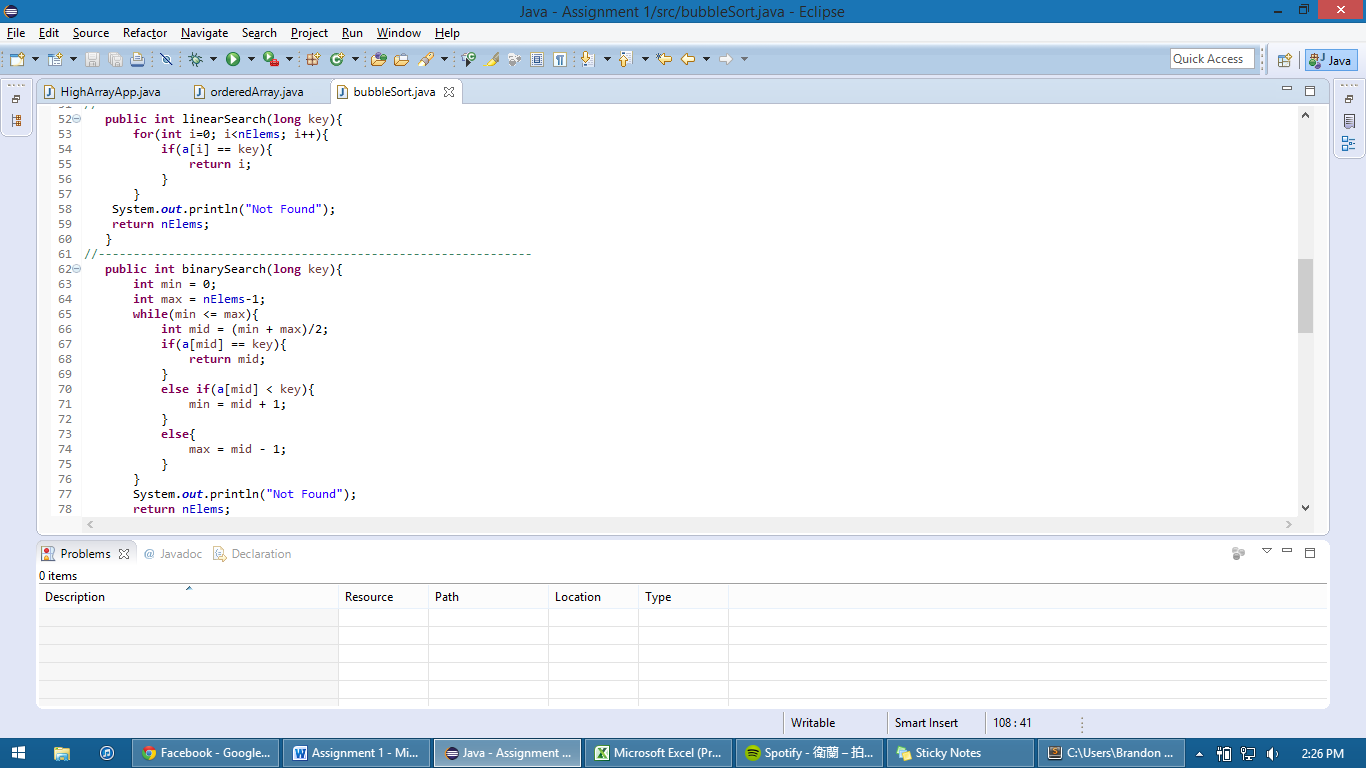


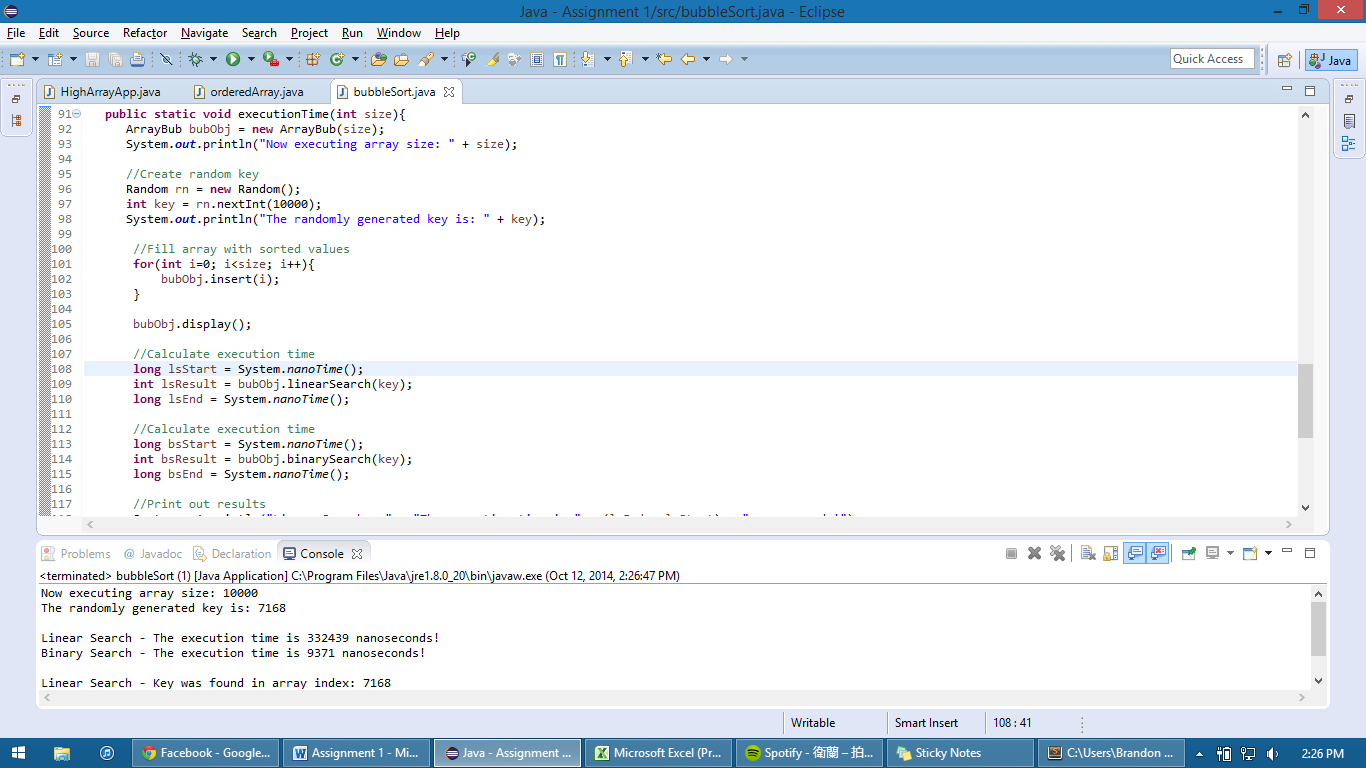
2.4 Screenshots





Q5 Screenshots





Execution Time Table:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Execution Time (ns)** | |
|  |  | Linear Search | Binary Search |
| 10,000 Integers | 1st | 332439 | 9371 |
|  | 2nd | 102186 | 17403 |
|  | 3rd | 378401 | 9817 |
|  | 4th | 408744 | 17403 |
|  | 5th | 66934 | 18742 |
| 100,000 Integers | 1st | 98170 | 10709 |
|  | 2nd | 411868 | 11155 |
|  | 3rd | 182061 | 11156 |
|  | 4th | 157965 | 10710 |
|  | 5th | 49978 | 11156 |
| 1,000,000 Integers | 1st | 326638 | 22757 |

As displayed by the table above, binary search is far superior to linear search in every instance. It is clear that what we learned in class aligns with the data obtained and displayed above. Binary search is indeed very fast with O(lg(n)) and linear search is indeed slower with O(n).