**ITWORKS**

**Data Structures and Algorithms – Project Documentation**

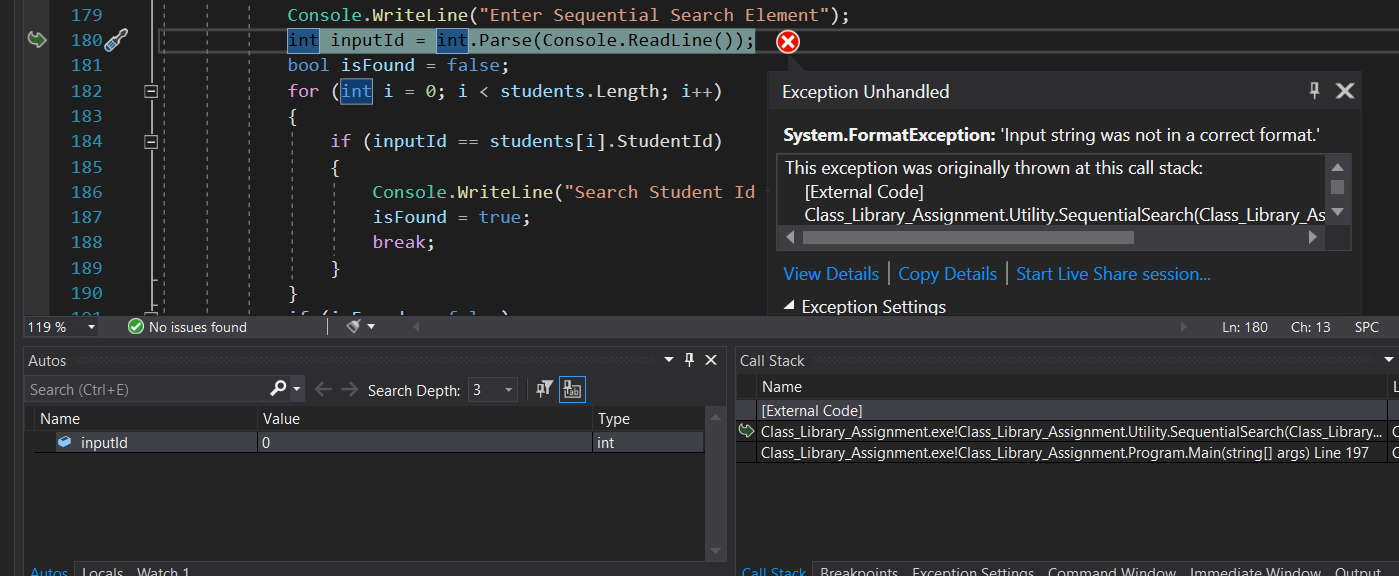
1. **GitHub Repository Link details and Credentials**
2. **Verification and Validation (Recorded during an oral feedback session with developer)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Module** | **Iteration** | **Project Lead Feedback** | **Developer Response** |
| **Sorting** |  | **Needs to implement CompareTo()** | **Needs to implement CompareTo() StudentID** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **Sorting Algorithms – Evaluation**

|  |  |  |
| --- | --- | --- |
| **Name of Sorting Algorithm** | **Advantages** | **Disadvantages** |
| **Bubble Sort** | **In-place, no external memory is needed.** | **Not so good sorting algo in general as it requires many comparisons** |
| **Selection Sort** | **Selection sort algorithm is 60% more efficient than bubble sort algorithm.** | **Running time of Selection sort algorithm is very poor of 0 (n2).** |
| **Quick Sort** | **Fast and efficient** | **Quick sort is undoubtedly a fast and efficient sorting algorithm, but when it comes to stability** |

1. **Error Report** (copy and paste screen shots of errors during runtime and debugging)

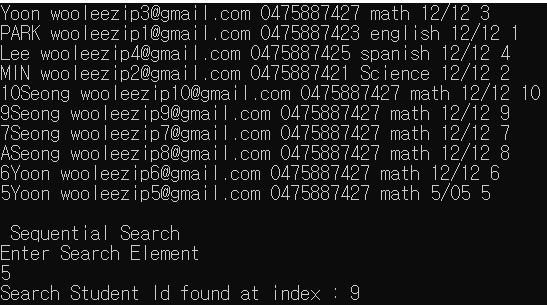


1. **Testing**
2. **Test Plan Template**

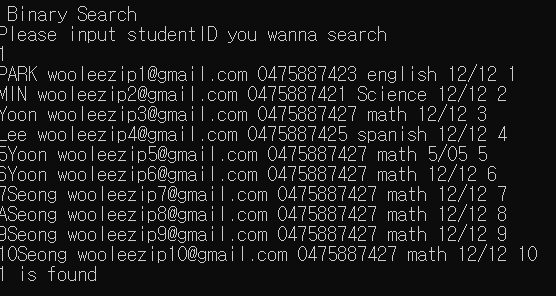
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Algorithm** | **photo** | **Expected output** | **Actual Output** | **Pass/Fail** | **Comments** |
| **Sequential Search** |  | **Found** | **Found** | **Pass** | **When number 1 is found Pass** |
| **Binary Serarch** |  |  |  | **Pass** |  |
| **Bubble \_ ASC** |  |  |  | **Pass** |  |

1. **Screen shots - Testing Searching and Sorting Algorithms**

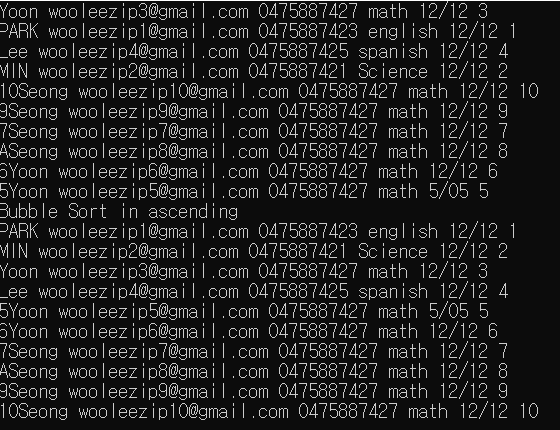
**// Sequential Search**

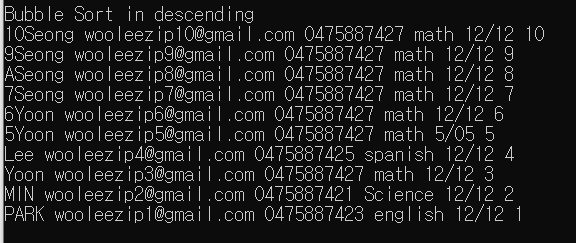


**// Binary Search**



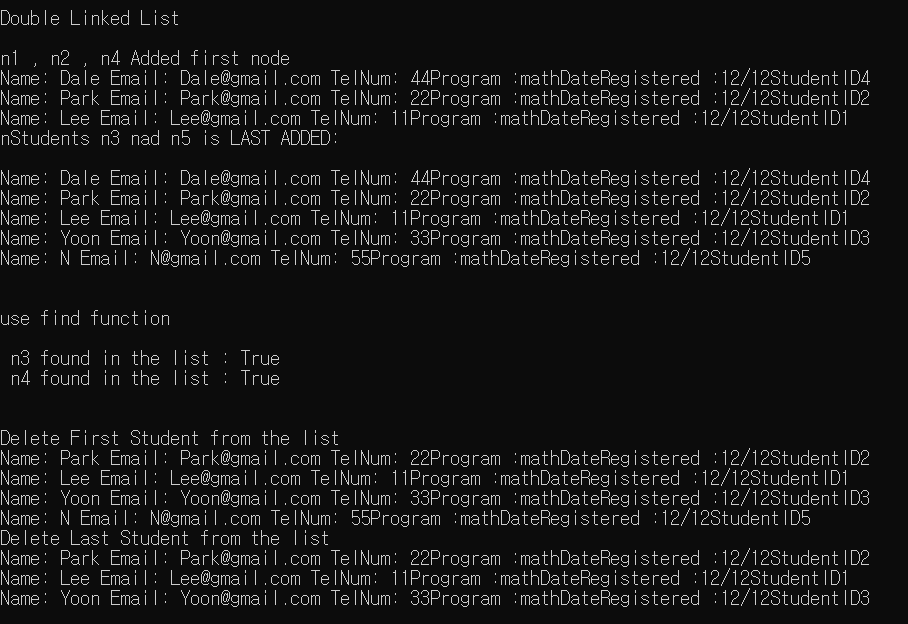
**// Bubble Sorting Ascending and Descending**



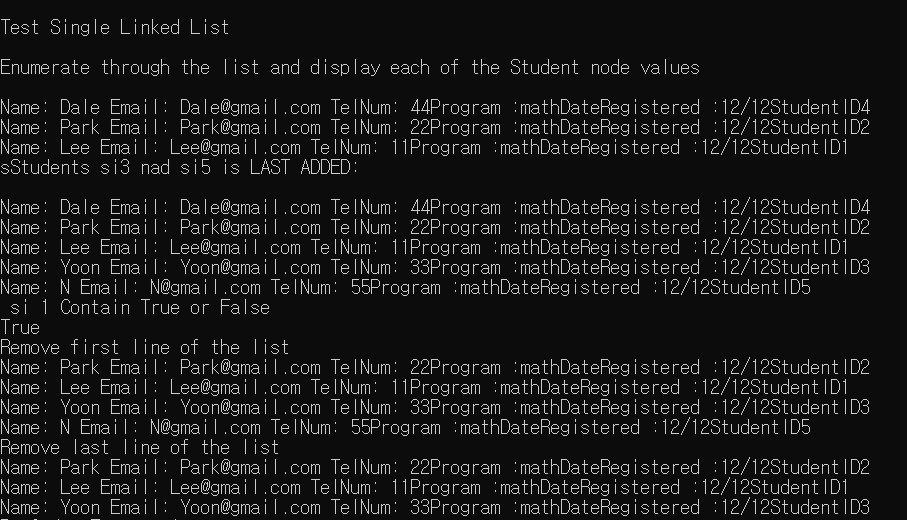


1. **Screen shots – Testing Single and Double Linked Lists Algorithms**

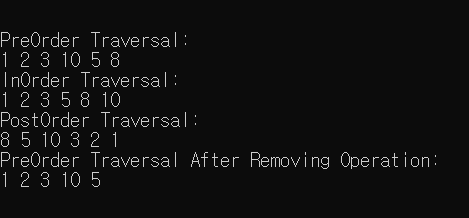
**// Test Double Linked List**



**// Test Single Linked List**



1. **Screen shots – Testing Binary Tree Algorithms**



1. **Further Research Report**

|  |  |
| --- | --- |
| **Language** | **Suitability**  Provide at least 2 reasons why it would be suitable for this type application |
| C++ | Data Structure Utilities and MVC (GUI) Applications: C++ is object – oriented:  When speed is a critical metric, C++ is the most preferred choice. The compilation and execution time of a C++ program is much faster than most general-purpose programming languages.  C++ is also used to develop GUI-based and desktop applications. Most of the applications from Adobe such as Photoshop, Illustrator, etc. are developed using C++.  The C++ Standard Template Library (STL) has many functions available to help write code quickly. For example, there are STLs for various containers like hash tables, maps, sets, etc. |
| C# | Data Structure Utilities and MVC (GUI) Applications:  While C++ is the fastest in terms of speed, c# is also not bad in speed aspect, So it would be good for data structure utilities.  C# is a programming language that is remarkably scalable and easy to maintain. Because of the strict nature of how static codes must be written, C# programs are reliably consistent, which makes them much easier to adjust and maintain than programs that are written using other languages. C# is completely object-oriented, which is a rare characteristic for a programming language. |
| VB.NET | Data Structure Utilities and MVC (GUI) Applications:  While VB.NET wouldn’t be the best in terms of speed, however it is easy language to use for data structure utilities.  Object-Oriented – VB.Net is all an object. From primitive data types to user-defined data types, each and everything is object-based. It easily gives all the advantages of any OOP’s based language like Java etc. It is a highly structured language. It is self capable of developing efficient applications. |

1. The organization is keen on moving to a more Agile approach when developing larger scale applications as they feel the current ad-hoc approach would be inadequate. Your experience and research has convinced you that the SCRUM approach will be best suited for the company’s future large scale project management/implementation. Outline four reasons (100 words) why agile techniques are suitability for the development large scale application

Agile follows an iterative process where projects are divided into sprints of a shorter span. Unlike the traditional approach, less time is spent on upfront planning and prioritization as agile is more flexible in changes and specifications developments. When it comes to making changes in the product or a process, agile methodology is much more flexible than the waterfall methodology. While working, if team members feel that there is a need to experiment and try something different than planned, the agile methodology easily allows them to do so. The best thing about this methodology is that it focuses more on the product than following a rigid structure.

Unlike the traditional approach, agile methodology isn’t linear or follows a top-down approach. This way, all the last-minute changes can be accommodated without affecting the result and disrupting the project schedule.

This is the best methodology to follow in case of complex projects. A complex project may have various interconnected phases and each stage may be dependent on many others rather than a single one as in simple projects. So, Agile methods are preferred for large and complex projects.

Agile divides a project into parts (called iterations) where each release is sent to the customer after every single iteration. Additionally, the success of the project can be easily foreseen through the success of these iterations. This removes the need for upfront planning completely.