**Final Programming Languages 3 Documentation**

**Contact Information**: Prithviraj Kalburgi, [prithviraj.kalburgi@tuni.fi](mailto:prithviraj.kalburgi@tuni.fi)

**Project point worth**: 4/5 points

**Design Solution**: This project is designed in two parts. The first part of the project contains the main solution which is the user interface and function calls for all different 5 choices provided in the menu list. The second part of the project contains the function codes which are fetched from the function calls. The programmed is designed in a way where “Array List” and “Values” are set within a defined boundary. If a user inputs any value outside the boundary, the program will not run.

**Testing**:

Phase 1

1)

Text

Description automatically generated

Text

Description automatically generated2)

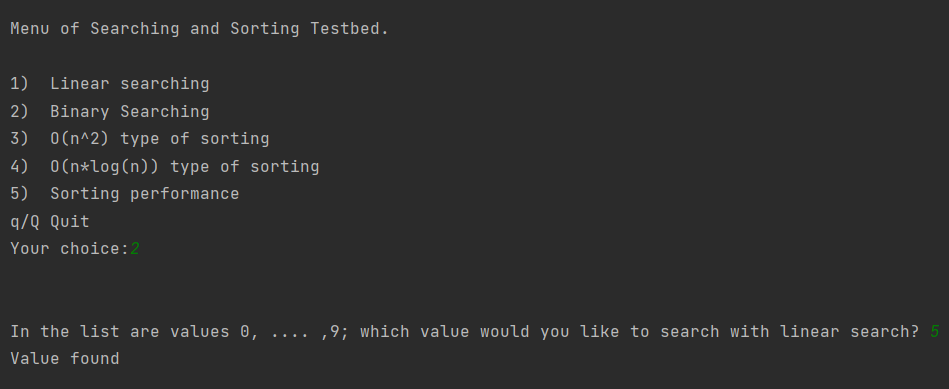
In this phase the user interface is displayed with a menu list from 1 to 5 with different functions. In picture 1, the “Linear searching” function is tested and user input is 4 which is set within the values bracket, therefore the result was “found”. In picture 2, the same function was tested but this time the user input was 12 which is outside the value bracket so therefore the result was “not found”. This phase would give me 1 point.

Text

Description automatically generated3)

Here the “Q” command was tested to stop the program and in bottom a different command for example as per the screenshot, “Q1” was tested, and it gave a output of “No option found”.

Phase 2

4)

5)

Text

Description automatically generated

In this phase the “Binary Searching” function is tested. Very similar to “Linear searching” and the output in picture 4 and 5 are similar to picture 1 and 2. This phase would give me 2 points.

Phase 3

Text

Description automatically generated6)

In this phase the “Insertion sorting” function is tested. It prints out a data set of 10 random values within an array and uses the “Insertion sorting” function to sort out the values. This phase would give me 3 points.

Phase 4

Text

Description automatically generated7)

In this phase the “Quick sorting” function is tested. It is a very similar function to the “Insertion sorting” function and has similar output as can be seen in picture 6. This phase would give me 4 points.

**Workhours**:

* 9/12/22, Approx. 3 hours, went through the slides given by the teacher and did some self study so I can be prepared to start this project. Read through the project details and instructions and watched several videos to understand the overall goal of the project.
* 10/12/22, 2 hours, copied all function code from slides and put them into separate files. Started working on the final UI program for Linear and Binary searching.
* 11/12/22, 2 hours, Worked on Insertion and Quick sorting functions.
* 12/12/22 2 hours, Testing/debugging code. Finalizing project and completing documentation.

**Instructions on how program can be tested**:

This project is in two parts so therefore the program can be tested very easily. The main program and the functions program are in the same source folder so everything should work smoothly.