**Programming Project Report**

Name: Blake Williams

Date: 3/3/2023

**Academic Integrity Statement:** I pledge that I have neither given nor received unauthorized help on this programming assignment.

**Problem Statement:**

During this homework the overall goal was to design code that was able to read in multiple txt files then parse through them utilizing recursion to find specific passwords, or hashes. The inputs of this code were when a user wants to find a specific password they must input the name of the password they are looking for. Then for finding specific hashes, the code reads from a txt file and parses through the txt file and compares it to the ordered list of hashes to find a match. The outputs are printing to the screen the results of the running through the information.

**Design:**

To begin the code it was chosen to create a password class that would store all the information of a password so later on in the code it could be possible to utilize the getters to get certain attributes of the Password class. After this it was chosen to utilize binary searching to parse through all the information presented in the txt files. The main algorithm chosen was recursive binary search which is able to call upon itself utilizing recursion for optimal performance. Other algorithms used were iterative binary searching which does not use recursion, but is still as fast. Finally linear searching was implemented which was the slowest of them all.

**Implementation:**

The starter code consisted of reading the given txt files into specific vectors. These vectors would be what are used to implement the algorithms mentioned earlier. The use of function came into play when designing the code, as it would held to consolidate specific algorithms to specific locations and so they could be utilized later in the main function. First the Password.H file and Password.cpp files were generated then after that progress began on the main function. In the main function a switch statement was implemented with simple error checking to prevent a user from entering a number other than an integer. Then outside of main skeleton functions were generated that would house each menu function. Progress began with the recursive binary search. After recursive binary was successfully implemented the next menu was created. Once all of the menu functions were completed then they were added to their respective case locations.

**Testing:**

Testing each menu function consisted of doing the very end of the list then the very beginning to check and make sure the calculations for the binary searches were operating correctly. Then for the print function that prints between two passwords, the order of the function were checked as well. So if the first input is later in the list and the second is earlier then it will still print out in order. A sample input output would look like the following.

// Start of the sample output

Welcome to the Password database

1)Search recursively two passwords and pull all the passwords between them

2)Search recursively by password

3)Search recursively of given file by hash

4)Compare linear and binary search

5)Iterative binary search for password

6)Exit system

1

Please input 2 passwords you wish to search, then this will display the passwords and print out the passwords between each of them

thecat

Please input the second password you wish to search for

thanks

Password:thecat

Rank: 3123

Hash: 9b65d1

The value has been found and has gone through 12 number of itterations using Bianary Search

Password:thanks

Rank: 1664

Hash: 71d3e8

The value has been found and has gone through 11 number of itterations using Bianary Search

Printing all data between two variables

Password:thanks

Rank: 1664

Hash: 71d3e8

Password:theater

Rank: 4471

Hash: 582726

Password:theatre

Rank: 5264

Hash: a0ff69

Password:theboss

Rank: 3014

Hash: b248e0

Password:thecat

Rank: 3123

Hash: 9b65d1

Welcome to the Password database

1)Search recursively two passwords and pull all the passwords between them

2)Search recursively by password

3)Search recursively of given file by hash

4)Compare linear and binary search

5)Iterative binary search for password

6)Exit system

//End of the sample output

**Conclusions:**

Overall this project was a success as the code was able to read and print the desired information to the terminal. If this project were to be continued multiple different functions would to test out other methods of parsing through information. This project took about 7hrs to fully complete including this report.