

Data Structure

Fall 2020, programming Assignment #2 Note

Due Date: 17:00 on Wednesday, January 13th, 2021

Submission: Place all files in a folder and zip it. Both the zip file and the folder inside should be named with your student ID (e.g., B123456789.zip with a folder named B123456789 in it). **Do not create additional folders or sub folders.** Upload the zip file via [CEIBA](#) (to Programming Assignment 2). The folder should contain the following:

- 1) **Source code files:** You can use the programming language of your choice. Please include all the source code needed to run / compile your program.
- 2) **Readme.pdf :** The file named “Readme.pdf” is a simple report that describes how your solution is implemented (e.g., data structure, algorithm, ...). Optionally, you can include simple explanations on how your code is structured. This file should not exceed two pages.
- 3) **Dictionary.txt :** This is the dictionary file generated by your program with the [password file](#) as input. The content of the file is described in [Programming Assignment #2](#). Some more details about the file is as following
 - 1) The three columns should be separated by a single white space.
 - 2) The 2nd column (representations the salt) should always contain three digits. Hence you need to pad the numbers with leading 0’s if they have only 1 or 2 digits.
 - 3) The 3rd column (representing the hash value) should not be padded with leading 0’s.
 - 4) The order of the passwords in your dictionary file should be the same as the order in the input password file. The salt should be from 000 to 999. The first 5 lines of the correct dictionary file are already given in page 2 of [Programming Assignment #2](#). So please double check to make sure the first 5 lines of your dictionary.txt file match those.
- 4) **results_pa2.txt :** A list of hash values is provided in [list_pa2.txt](#). You need to run your program with the [password file](#) as input, then query your program with each hash values in [list_pa2.txt](#). The results are then stored in the file named result_pa2.txt. There is no requirement to generate this file automatically. It is perfectly okay for your to manually run each search and cut-n-paste the output to the result_pa2.txt file (as long as you don’t find the manual work too tedious). Following are the details about the file format:
 - 1) The file contain 4 columns separated by a single white space character.

- 2) The 1st column is the hash value to be searched. This number should not be padded with leading 0's.
- 3) The 2nd column is the password recovered. If no password is found, use "*****". So this column should contain exactly 6 characters.
- 4) The 3rd column is the salt corresponding to the hash value and should always contain 3 characters. Hence you need to pad the numbers with leading 0's if the salt < 100. If no password is found, use "***".
- 5) The last column is the number of entries in the dictionary file that have been searched for this hash value. This number does not need to be padded with leading 0's.

Please note that the format listed above is slightly different from the command line output specified in [Programming Assignment #2](#). So if you intend to generate this file from your program automatically, please take care to reformat the output so that they match our requirement here.

***Important Note:** *When searching the dictionary file for the given hash value, you are only expected to search the file sequentially. Therefore the number of entries searched for any given hash value is fixed. In fact, we will use this information to evaluate the correctness of your program. So please refrain from using special search algorithms. Unless you can still manage to produce the expected output for the last column, in which case your results will be considered correct.*

- 5) **Screenshots.pdf** : In order to verify your program has the functionalities required, please provide the screen shot(s) of your command line interface. In the screen shot, we expect to see the name of the password file being passed to the program, followed by three searches. Please make sure your screen shot(s) clearly shows the hash values being given to your program, as well as the output generated. You can pick any three hash values for the search. Please include exactly one "failed" search where the input is a hash value not found in the dictionary file.

If you have any questions regarding the requirement specified in this file or any of the materials related to this assignment listed at the course website, please do not hesitate to contact any one of the three TA's via emails:

王璿	d07921012@ntu.edu.tw
王思涵	stellawang18@gmail.com
余若萱	q8611027@gmail.com