COURSE NAME: Algorithm Analysis

STUDENT: Ertuğrul ŞENTÜRK

HOMEWORK SUBJECT: Usage of Hash Algorithm

ALGORITHM:

1. Load factor and hash table were taken as input from the database, if file does not exist an empty hash table was created. Load factor value is taken as 0.
2. An input taken from the user for method.
3. For input 1 we get a file name from the user and read it.
4. If the word exists in the hash table, the name of the document is added to its struct, otherwise a new word is added to the first empty index that results from the hash value.
5. While adding the word, how many steps it taken and the current table occupancy rate were printed to the user.
6. For input 2 we get a word from the user.
7. The word was searched in the hash table, if any, the documents containing the word, if not, the user was informed that the word could not be found.
8. We printed how many steps to search it taken.
9. For input 3 the hash table is cleared. The load factor is reset, and the empty table values are saved in the database file.
10. For input 3 the text on the screen has been cleared.
11. For input Q memory deallocated and program terminated.

Complexity:

1. If we assume the word count in the file is n, finding a word's hash value and putting into the table has in theory O(1) complexity. For all words it has O(n) complexity.

2. Finding a word in the hash table also has in theory O(1) complexity.

Screenshots:

```
       Welcome to the Word Search Program
App Description:
       - Application fills a hash table with input words and prints its information.
       - Input can be get from a file or keyboard.

Please select an operation:
1. Add a File Into Hash Table
2. Search a Word in Hash Table
3. Clear Hash Database
4. Clear Screen
Q. Quit
Input:
```

Welcome Screen

```
Old database is not exists creating new one..

        Welcome to the Word Search Program
App Description:
        - Application fills a hash table with input words and prints
        - Input can be get from a file or keyboard.

Please select an operation:
1. Add a File Into Hash Table
2. Search a Word in Hash Table
3. Clear Hash Database
4. Clear Screen
Q. Quit
Input: 1
File reading option selected
Please enter the file name: doc1.txt
- Lorem added in 1 steps - Load Factor: 0.001
- ipsum added in 1 steps - Load Factor: 0.002
- dolor added in 1 steps - Load Factor: 0.003
- sit added in 1 steps - Load Factor: 0.004
- amet added in 1 steps - Load Factor: 0.005
- consectetur added in 1 steps - Load Factor: 0.006
```

A dding a New File

```
Input: 2
Word search option selected
Please enter the key-word: a
- a founded with 1 steps in doc1.txt, doc2.txt, doc3.txt, doc4.txt files.
```

```
Input: 2
Word search option selected
Please enter the key-word: nor
- nor founded with 6 steps in doc3.txt files.
```

```
Input: 2
Word search option selected
Please enter the key-word: test
- test couldn't found. Process operated for 997 steps.
```

Search Samples

```
Please select an operation:
1. Add a File Into Hash Table
2. Search a Word in Hash Table
3. Clear Hash Database
4. Clear Screen
Q. Quit
Input: 3
Database successfully cleared.
```

Clearing Database

```
- four added in 5 steps - Load Factor: 0.794
- promising added in 3 steps - Load Factor: 0.795
- together added in 2 steps - Load Factor: 0.796
- Forget added in 4 steps - Load Factor: 0.797
- bored added in 3 steps - Load Factor: 0.798
- library added in 7 steps - Load Factor: 0.799
- web added in 5 steps - Table is almost full - Load Factor: 0.800
- design added in 2 steps - Table is almost full - Load Factor: 0.801
- classes added in 1 steps - Table is almost full - Load Factor: 0.802
- exaggerating added in 3 steps - Table is almost full - Load Factor: 0.803
- convenience added in 3 steps - Table is almost full - Load Factor: 0.804
```

Table Almost Full

```
Table Full: Failed to add by.
Table Full: Failed to add days.
Table Full: Failed to add equals.
Table Full: Failed to add about.
Table Full: Failed to add characters.
Table Full: Failed to add per.
```

Table is Full