**CS 307 FALL 2021-2022 PA5**

**ELIF CEMRE DURGUT – 26493**

Pseudocode of my algorithm: (I put indentation lines to make it clearer.) It works for more than 1 depth with subdirectories and files.

**listFiles function(path, database):**



open the directory

while there is a directory:

if it is a regular file by checking stat():

if it is a txt file by checking d\_name and not the database.txt in root directory:

open the file

while there is a word in the file:

compare the word with the names in database

if there is a match:

go backward with **fseek**, correct the Mr./Ms. with **fputs**

go forward with **fseek**, correct the last name with **fputs**

close the file

update the path to go deeper

call listFiles with updated path (recursively)

close the directory

**main part:**

get the current directory to get the path

open database.txt file

count the number of lines in it to form an array

move the file pointer to the beginning with **fseek**

read the txt line by line and add f/m, name, and last names to the array

close the file

call listFiles with path and database array

**Step by step explanation:**

1. In **main** part, I first access the current working directory to read the database.txt.
2. After I open the txt, I read it line by line and store the number of lines in it.
3. Using the calculated int above, I declare an array and move the pointer to the beginning of the txt file.
4. I again read the file line by line and store the f/m, name and last name in char array like [“f”, “elif”, “durgut”]
5. I close the file
6. I call the listFiles function with the database array and current working directory path.
7. Inside **listFiles** function, after I opened the directory, I first check if it is file or not. Because it may be a directory with a name like “a.txt”.
8. After making sure that it is regular file by utilizing stat(), I check whether it is a txt file or not.
9. If it is a txt file, I open and read it word by word, and search for any name in the database.
10. If I find a match, I go strlen(word) + 4 units backward, 4 comes from “Ms.” or “Mr.” (3) and a space (1). I modify the Mr/Ms string based on the data in the database.
11. After that, I move strlen(word)+2 units forward. 2 comes from the space before and after the name. And I correct the last name.
12. I close the file
13. I update the path and call listFiles function recursively.