CSE142 – Spring 2017

Homework #2

Due: 04.05.2017

In this homework, you have to build a simple address book using C programming language. The user will give the necessary information and your program has to save and keep that information in a text file.

Your program should be menu driven and have 4 abilities below:

- List all records.
  Program will display all the records.
- Update a record.
  User can update a record's phone number or address.
- Create a new record.
  User can create a new record.
- Delete a record.
  User can delete a record with using its id.

All records will have fields as **first name**, **last name**, **phone**, **address** and **id**. You can use structs to construct a record, for example :

```
struct recordData {
    int id;
    char firstName[15];
    char lastName[15];
    long double phoneNum;
    char adress[100];
};
```

When **updating or deleting** a record, your program firstly has to take the **id** as input and check if this id exists in the address book. If it exists, then your program can update or delete the specified record.

When **creating** a new record, your program firstly has to take the **id** as input to check if this id exists in the address book. If it does not exist, then your program has to take first name, last name, phone and address as input and create the new record.

The address you will get from user will be in District/Province format with **no** space character in between. The name and surname fields will also **NOT** contain any space character.

Sample runs of the program are as follows:

(Your program's output is given in black. User input is in red.)

```
ADDRESS BOOK
1 - List all records
2 - Update a record
3 - Create a new record
4 - Delete a record
5 - Exit
Enter your choice : 1

No      First Name  Last Name   Phone Num   Address
==      ==========  =========   =========   =======
1       ali         yılmaz      324324444   Kadıköy/İSTANBUL
2       ugur        esgin       3245435455  İçerenköy/İstanbul

ADDRESS BOOK
1 - List all records
2 - Update a record
3 - Create a new record
4 - Delete a record
5 - Exit
Enter your choice : 2

Please enter the record number of the person you want to update : 2

2       ugur        esgin       3245435455  İçerenköy/İstanbul

1 - Update phone number
2 - Update address
Enter your choice : 1
Enter the new phone number : 5359998989
2       ugur        esgin       5359998989  İçerenköy/İstanbul

ADDRESS BOOK
1 - List all records
2 - Update a record
3 - Create a new record
4 - Delete a record
5 - Exit
Enter your choice : 3

Enter the record number to create a new record : 3
```

```
Enter the first name : ziya
Enter the last name : çelik
Enter the phone number : 023443244
Enter the adress : Göztepe/İstanbul

ADDRESS BOOK
1 - List all records
2 - Update a record
3 - Create a new record
4 - Delete a record
5 - Exit
Enter your choice : 1

No      First Name  Last Name   Phone Num   Address
==      ==========  =========   =========   =======
1       ali         yılmaz      324324444   Kadıköy/İSTANBUL
2       ugur        esgin       3245435455  İçerenköy/İstanbul
3       ziya        çelik       023443244   Göztepe/İstanbul

ADDRESS BOOK
1 - List all records
2 - Update a record
3 - Create a new record
4 - Delete a record
5 – Exit
Enter your choice : 4

Enter the record number to delete :  1
1       ali         yılmaz      324324444   Kadıköy/İSTANBUL
Deleted.

ADDRESS BOOK
1 - List all records
2 - Update a record
3 - Create a new record
4 - Delete a record
5 – Exit
Enter your choice : 5

Bye...
```

BONUS: Display all the records **sorted** alphabetically according to last name of the records.

Error Check:

- You are expected to create the text file if it does not exist.
- You have to make sure that your program does not crashes if the user enters an invalid input (e.g., entering integer for name, entering char/string for id).

- Your program should print an error message if user wants to delete or update a nonexistent record.
- Your program should print an error message if user wants to add a record with an existing id.