Project Report

Introduction:

The purpose of the project is to create a phone directory, which include all the data of each person.

There are 8 functions in the code which executes certain commands. Some functions are related to the file that is given first, which includes all the data and information of each person in the telephone book, other functions execute specific commands that are given by the desire of the user.

1.Description

program(): An infinite loop is done by while (1) which includes all the function mentioned, in order to stop the loop, the user must choose quit function. After declaring a variable x, the user input a specific number from 1-8, which each executes a certain command. A nested if is used which does a certain command given by the number the user chose. If the user press number 8, which is quit function, a warning is displayed to the user that all the changes will be discarded. If the user press 1 the program will quit, any other number the program will still run.

Load(): is a function that loads the phone directory from a file by reading its name then loading each entry. The function loads data from file by declaring a pointer of type FILE then we read from the file by using the function fopen and setting the mode on “r” as shown in code 1.1.

```
FILE *f=fopen("Contact.txt","r");
```

Code 1.1

Then the program will check if the file is found and the pointer points to the file by using if(f!=NULL) if this condition is not satisfied the program will print “File not found", but if the file was successfully found the program will continue in order to load the file. In order to read all records in the file a while loop was used and its condition is !feof(f), feof() is a function that test for the end of the file, the feof function returns a nonzero value if the end-of-file indicator is set. Otherwise, it returns zero. While the condition is true, the program reads the string until a semi column is found by using "%[^,]," and using fscan() with three parameters: file pointer, format and the variable. Then puts this string in the first name of the record x[count] where x is an array of fields of type contact as shown in code 1.2.

```
fscanf(f,"%[^,],",x[count].first);
```

Code1.2

the program reads the string until a semi column is found and puts this string in the last name. The program reads the date of birth day, month, year, all initialized as integers, by using fscanf(). The program reads the address until a semi column is found and the phone number is also read until a semi column is found by fscanf() as shown in code 1.3

```
fscanf(f,"%[^,],",x[count].address);
fscanf(f,"%[^,],",x[count].number);
```

Code1.3

Finally, the email is read by fscanf() and with format %s. And with every record loaded the variable count is incremented by 1. This variable represents the number of records loaded. At the end the file is closed by command fclose() and the file pointer is given to this function fclose(f); .

<u>Query:</u> is a function that searches for a specific entry according to the person's last name. It asks the user to enter the last name of the required record to search for it. Firstly, an array of characters lastname and variables called flag and I are initialized. Then the user is asked to enter the last name which will be assigned to the array lastname. A for loop is used while I is less than count. If I is less than count the program will compare the array of characters lastname to the last names found in the array of fields x[i]. last by using strcasecmp(str1, str2) which takes two parameters if they are equal it returns 0 and returns < 0 if `str1` is less than `str2`; > 0 if `str1` is greater than as shown in code 1.4.

```
if(strcasecmp(x[i].last,lastname)==0)
```

Code 1.4

If the two strings are equal the flag will be equal 1 and the program will print the first name, last name, date of birth, address, phone number and email of the record the user searched for. If the record is not found the program will print "Search not found". This check is done by testing if the flag equals zero.

<u>Add</u>: Explanation: The add function is used to add a record into a file. Two integers "Z" and "flag=0" are declared, then a print function asks the user if he would like to add an entry and if the answer is yes enter 1 and if no then enter 2 and the answer is then stored in "Z". An IF function is used to test if "Z" is equal to 1 or not is yes then it executes the code used to add, if it isn't equal to 1 then the code is skipped, and the task isn't performed. If "Z" is equal to 1 then questions are prompted for the user to fill in. the first question is asked in a print asking for the last name and is stores as a string in the array x[count].last, “c” is the name of the array and count is the position in the array that was taken from the load function and it’s the last available position in the array and .last is to specify the type that was specified in the struct and is also used to specify the position in the array. This is repeated with all the other requirements. When it comes to the phone number and date of birth and the email there are validations, for the validations a do while loop is used in case the return from the validation function is 0 then the “flag” variable will be equal to 1 repeating the loop since the condition is “flag==1” and the flag is reset to 0 when the loop is repeated before anything else is executed.

For the date of birth validation “Valid_dob”,seen In *code 1.2*, three “d”,”m”,”y” variables are entered into the function and each is tested that they are within appropriate range using an IF function that returns 0 if they are outside that range. For the phone number validation “Valid_number”, seen in *code 1.1*, each element of number string is tested one by one using a for loop and that element is put inside an “isdigit” function that tests if element is a digit or not, if It is then 1 is returned if not then 0 is returned. For the email validation “Valid_email”,seen In *code 1.3*, the email string is put in a for loop ,with the condition that the I must be smaller then the length of the string so it knows where to stop, and each element is checked until the “@” is reached and the position of it in the string is stored as “x” and the position of the “.” Is stored as “y” then IF function is used to make sure that the “@” is before the “.” and that they are even there and if they are then 1 is returned.

```
Int valid_number(char phone[]);
```

Code 1.1

```
Int valid_dob(int d,int m,int y);
```

Code 1.2

```
Int valid_email(char em[]);
```

Code1.3

Delete: Explanation: This function deletes an unwanted record from a file. Two strings are declared “firstname” and “lastname”, then two prints are used to ask the user to enter the first and last name. The first name entered is stored in “firstname” and the second entry is stored in “lastname”. Then a for loop is used to go through the whole files comparing the two names entered with the names to match with a record in the file(as seen in the code below), this is done with an IF function with a string compare function, this IF function is part of a for loop where the condition is that "i" is less the count. When the entry is found then is overwritten by the record after is using another nested for loop that brings every record one step back overwriting the record wanting to be deleted. In the if statement if the record is found an int "c=0" that was declared is incremented because it is used in the end of the delete function in an IF ELSE function stating if "c=0" then an error will occur else a prompt saying that deletion is complete.

```
for(int i=0; i<count; i++)

  {

    if(strcasecmp(x[i].first,firstname)==0&&strcasecmp(x[i].last,lastname)==0)

    { c++;



 for(j=i; j<count-1; j++)

      {

        x[j]=x[j+1];


      }

      count--;

      i--;

    }
```

<u>Modify</u>: the modify function is built that one aspect of a contact can be changed at a time since otherwise deleting the contact and adding another one would be more efficient it.

The modify function asks the user which aspect of the contact (first name, last name, address, phone number, or date of birth) do they want to change and asks for the new data as an input from the user and replaces it.

The user will be asked for the last name and the following loop will determine the number of contacts with the same last name and the value will be saved in int variable v and will act accordingly.

```
for(j=0;j<count;j++)
 {
    if((strcasecmp(sample,x[j].last)==0))
      {
         rep[v]=j;
         v++;
      }
 }
```

in case only one user was found the program will ask the user which aspect needs to be changed and will change it.

In case of duplication in last names the program prints all the users with that last name and asks the user to enter the value corresponding to the user they want to modify its data.

In case the last name was not found the program will tell the user that the contact was not found, and the user will have to start over.

Print: the sort(print) function asks the user if they want to sort alphabetically or in ascending order by the date of birth and prints all the contact in the loaded file accordingly.

The function if asked to sort alphabetically will use bubble sort to sort all the users in the first names first the repeat the process for the last name second using stracasecmp()

Which compares which of the 2 strings comes first in the alphabetical order with ignoring upper and lower case differences it returns zero if the two strings are equal 1 if the first string comes first and -1 if the second string comes first.

```
if(strcasecmp(x[j].first,x[j+1].first)>0)
 if(strcasecmp(x[j].last,x[j+1].last)>0)
```

so that it can work in case of duplication of last name and thus the contacts with the same last name will be ordered according to the first name.

Save: The purpose of this function is to save the data that is given from the file after completing certain commands that are chosen by the user such as Add, Delete, and Modify back to the same file in a format similar to the one in the Load function.

Code: First step is to declare a variable of type FILE to represent the file, which will be a pointer of type FILE. In order to open the file, a function called fopen, known in the library, is used to open the file. When using fopen, the name of the file is given with the type of command you want, which in this case is “w” because save function writes data back into the file.

```
FILE *f=fopen("contact.txt","w");
```

After opening the file for writing, a for loop is used to write the data back into the file after changes done. The for loop stops if i<count. Count is a global variable that is used to know number of people in the file which is done in the load function. A function called fprintf is used in order to write into a text file. When writing back into the file the data is written in the same format again.

```
fprintf(f,"%s\n",x[i].email);
```

Finally, fclose function is used to close the file after processing the data.

```
fclose(f);
```

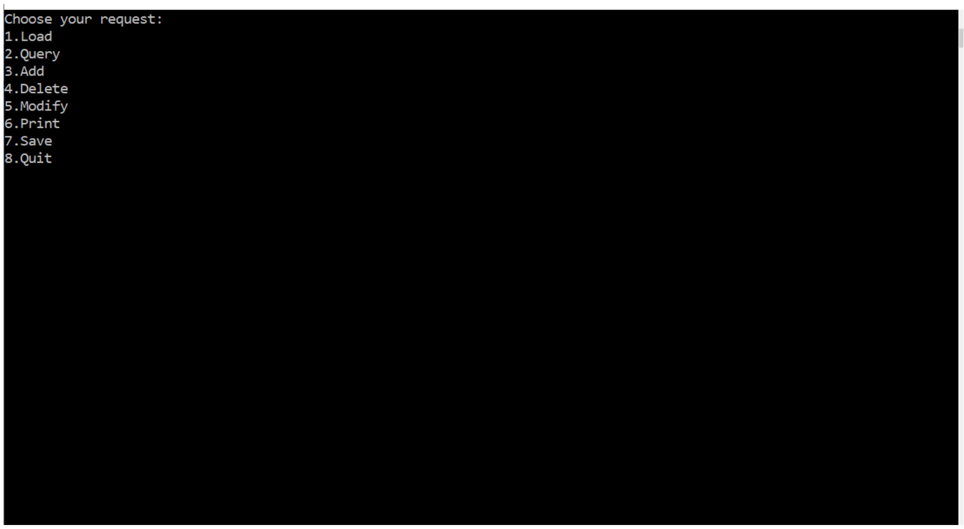
Quit: A simple function that exit the program when the user wants. If the user chooses to quit without saving, all the changes done will be discarded.

Code: function called exit(int), which terminates the process normally performing the regular cleanup for terminating programs, is used to quit the program. Zero is written inside exit function in order to a successful termination.

```
exit(0)
```

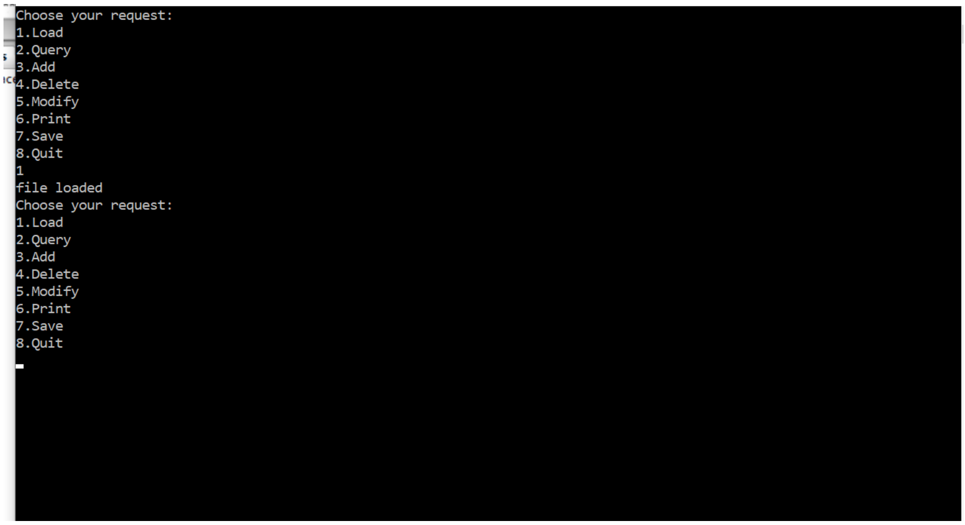
2.Sample runs:

1)Menu



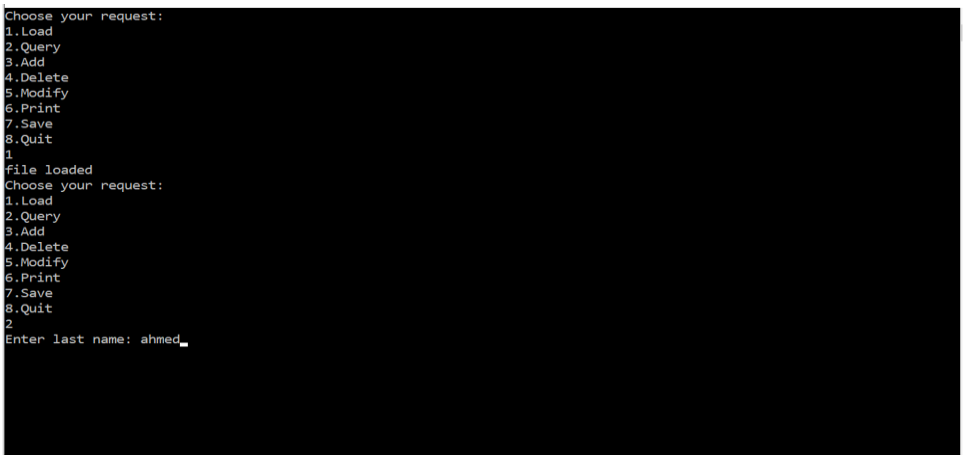
```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```

2)Load function

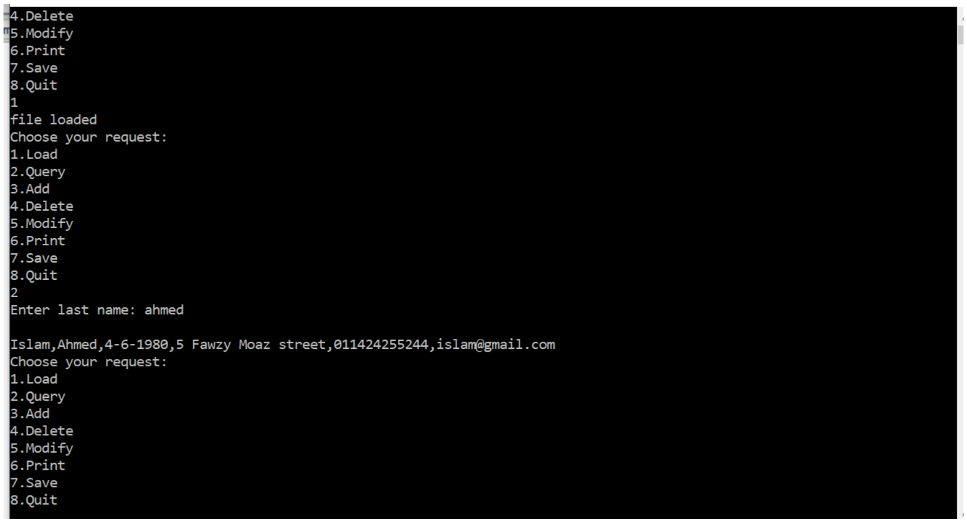


```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```

3) Query Function

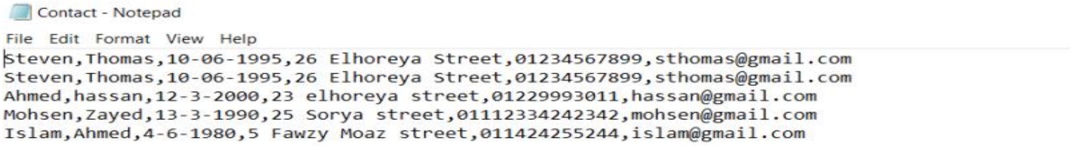


```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
2
Enter last name: ahmed
```

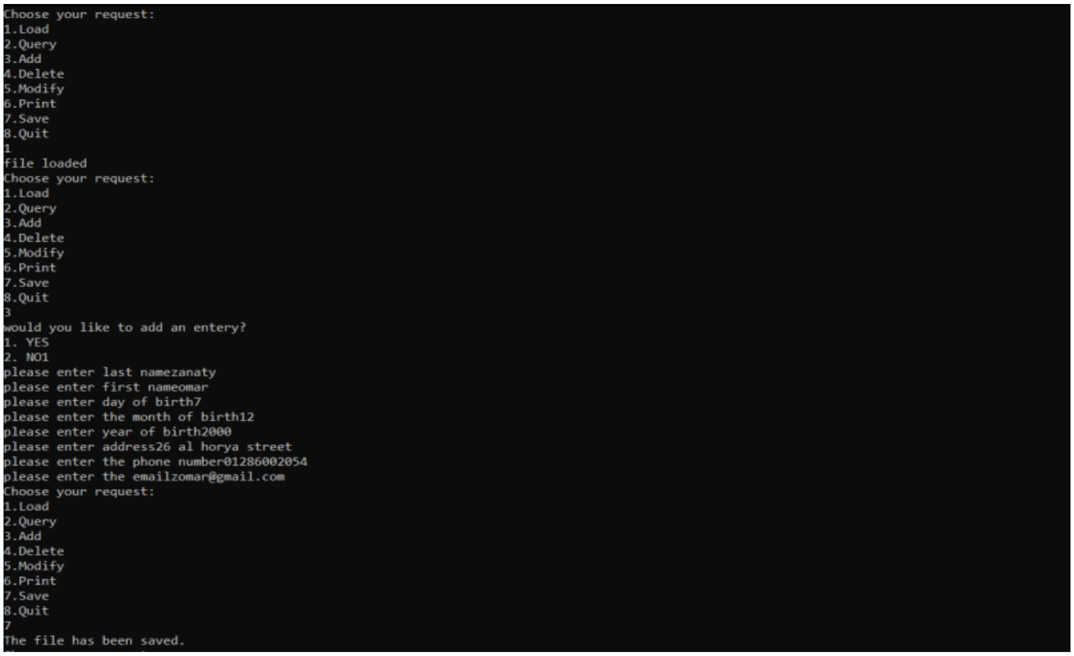


```
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
2
Enter last name: ahmed

Islam,Ahmed,4-6-1980,5 Fawzy Moaz street,011424255244,islam@gmail.com
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```

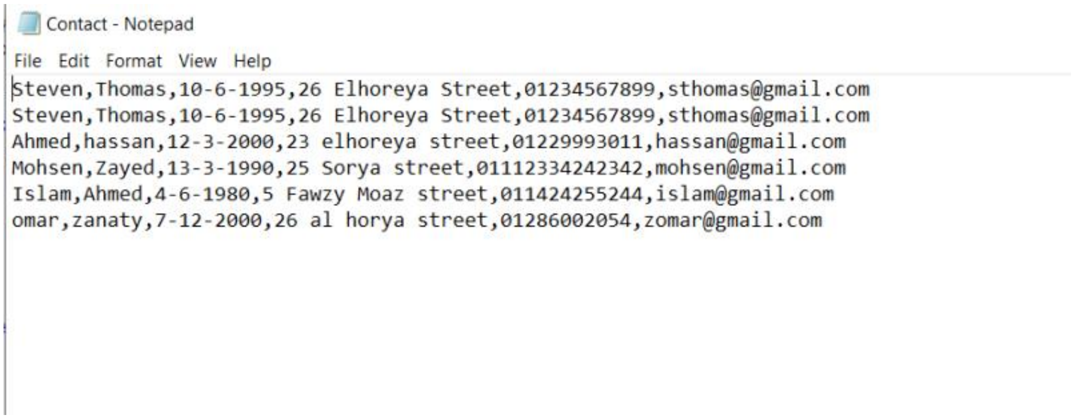
4) Add



```
Steven,Thomas,10-06-1995,26 Elhoreya Street,01234567899,sthomas@gmail.com
Steven,Thomas,10-06-1995,26 Elhoreya Street,01234567899,sthomas@gmail.com
Ahmed,hassan,12-3-2000,23 elhoreya street,01229993011,hassan@gmail.com
Mohsen,Zayed,13-3-1990,25 Sorya street,01112334242342,mohsen@gmail.com
Islam,Ahmed,4-6-1980,5 Fawzy Moaz street,011424255244,islam@gmail.com
```

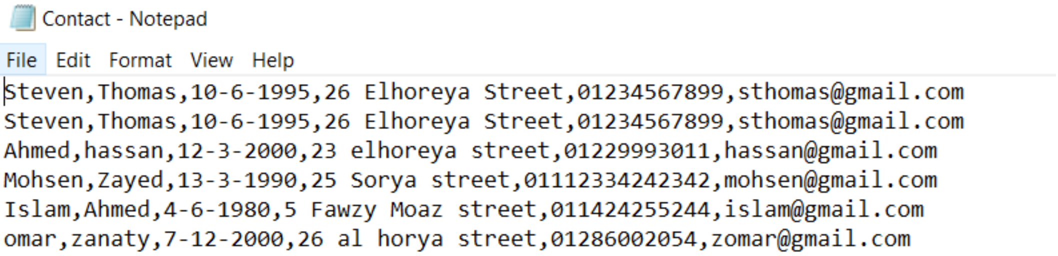


```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
3
would you like to add an entery?
1. YES
2. NO1
please enter last namezanaty
please enter first nameomar
please enter day of birth7
please enter the month of birth12
please enter year of birth2000
please enter address26 al horya street
please enter the phone number01286002054
please enter the emailzomar@gmail.com
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
7
The file has been saved.
```

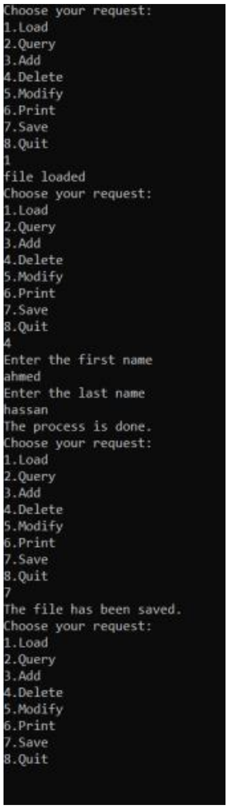


```
Steven,Thomas,10-6-1995,26 Elhoreya Street,01234567899,sthomas@gmail.com
Steven,Thomas,10-6-1995,26 Elhoreya Street,01234567899,sthomas@gmail.com
Ahmed,hassan,12-3-2000,23 elhoreya street,01229993011,hassan@gmail.com
Mohsen,Zayed,13-3-1990,25 Sorya street,01112334242342,mohsen@gmail.com
Islam,Ahmed,4-6-1980,5 Fawzy Moaz street,011424255244,islam@gmail.com
omar,zanaty,7-12-2000,26 al horya street,01286002054,zomar@gmail.com
```

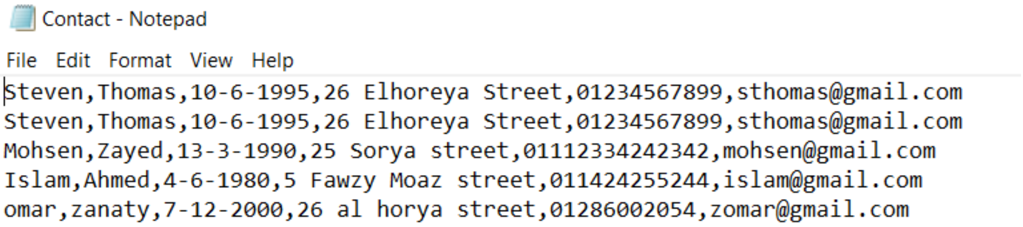
5) Delete



```
Steven,Thomas,10-6-1995,26 Elhoreya Street,01234567899,sthomas@gmail.com
Steven,Thomas,10-6-1995,26 Elhoreya Street,01234567899,sthomas@gmail.com
Ahmed,hassan,12-3-2000,23 elhoreya street,01229993011,hassan@gmail.com
Mohsen,Zayed,13-3-1990,25 Sorya street,01112334242342,mohsen@gmail.com
Islam,Ahmed,4-6-1980,5 Fawzy Moaz street,011424255244,islam@gmail.com
omar,zanaty,7-12-2000,26 al horya street,01286002054,zomar@gmail.com
```



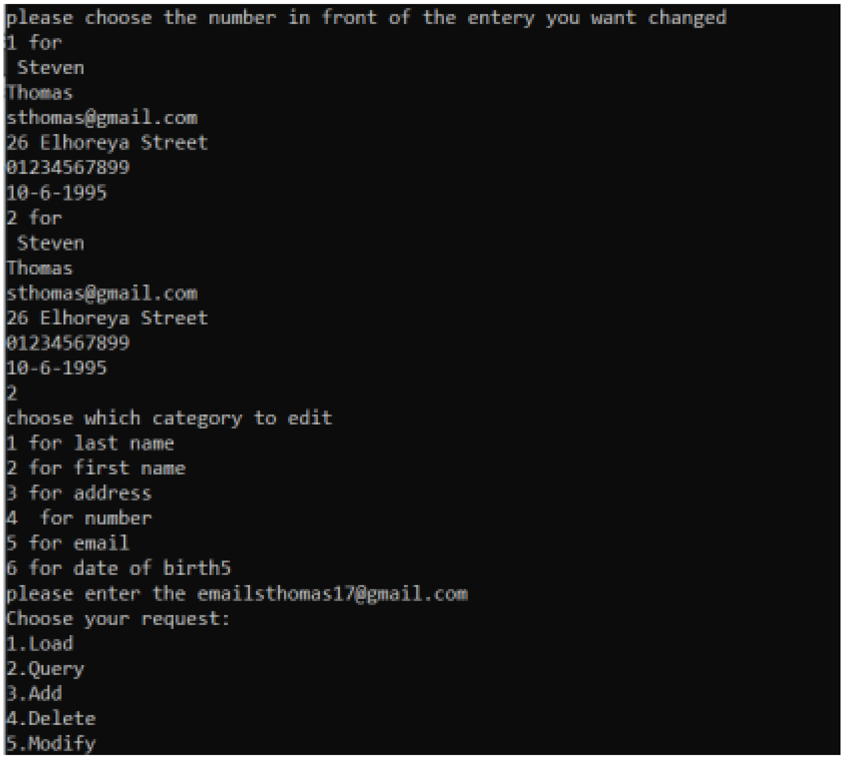
```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
4
Enter the first name
ahmed
Enter the last name
hassan
The process is done.
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
7
The file has been saved.
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```



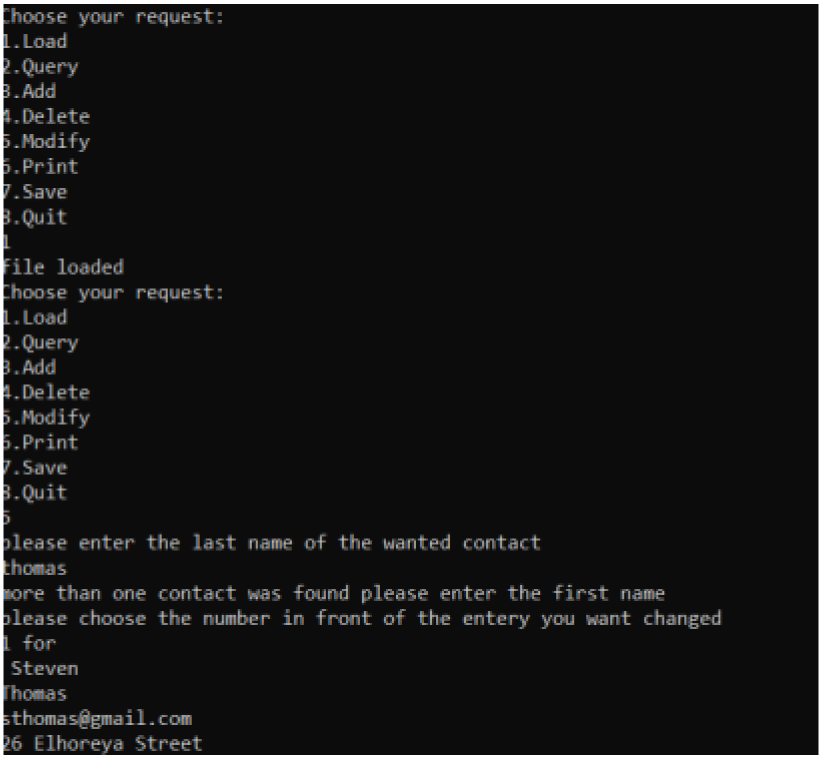
```
Steven,Thomas,10-6-1995,26 Elhoreya Street,01234567899,sthomas@gmail.com
Steven,Thomas,10-6-1995,26 Elhoreya Street,01234567899,sthomas@gmail.com
Mohsen,Zayed,13-3-1990,25 Sorya street,01112334242342,mohsen@gmail.com
Islam,Ahmed,4-6-1980,5 Fawzy Moaz street,011424255244,islam@gmail.com
omar,zanaty,7-12-2000,26 al horya street,01286002054,zomar@gmail.com
```

6) Modify

In this case there was duplication in the last name as the email of one of the needed to be modified.

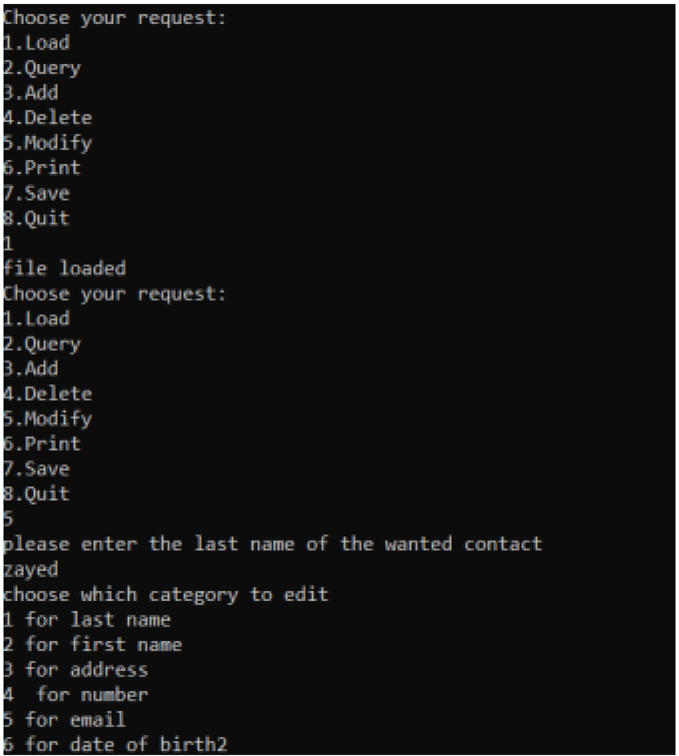


```
please choose the number in front of the entery you want changed
1 for
 Steven
Thomas
sthomas@gmail.com
26 Elhoreya Street
01234567899
10-6-1995
2 for
 Steven
Thomas
sthomas@gmail.com
26 Elhoreya Street
01234567899
10-6-1995
2
choose which category to edit
1 for last name
2 for first name
3 for address
4  for number
5 for email
6 for date of birth5
please enter the emailsthomas17@gmail.com
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
```

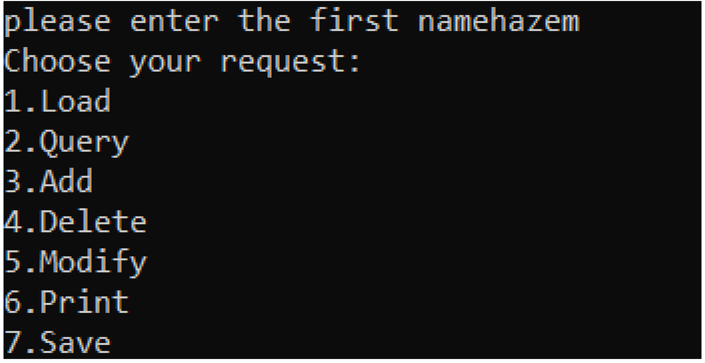


```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
5
please enter the last name of the wanted contact
thomas
more than one contact was found please enter the first name
please choose the number in front of the entery you want changed
1 for
 Steven
Thomas
sthomas@gmail.com
26 Elhoreya Street
```

While in this case no duplication was present, and the first name needed to be changed.



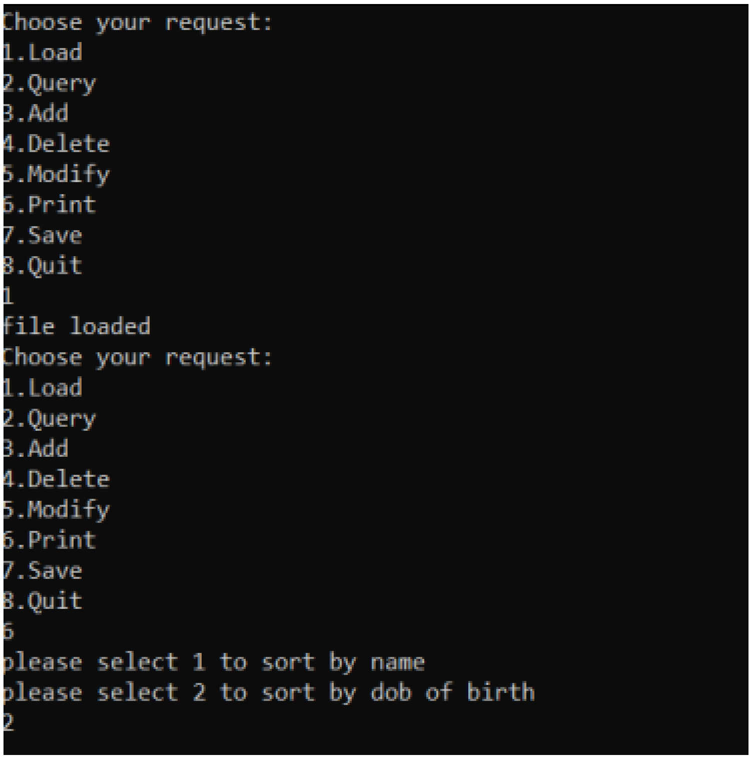
```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
5
please enter the last name of the wanted contact
zayed
choose which category to edit
1 for last name
2 for first name
3 for address
4  for number
5 for email
6 for date of birth2
```



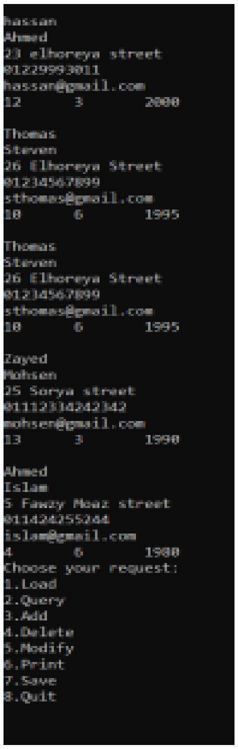
```
please enter the first namehazem
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
```

7) Print(sort)

In this case the user to sort according to the date of birth.

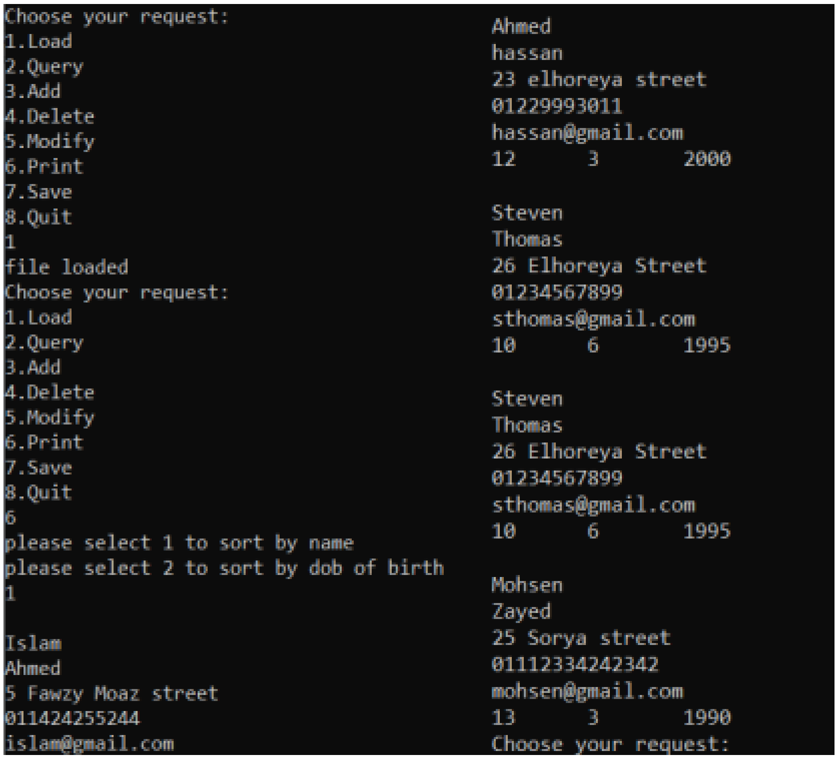


```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
6
please select 1 to sort by name
please select 2 to sort by dob of birth
2
```



```
hassan
Ahmed
23 elhoreya street
01229993011
hassan@gmail.com
12      3       2000

Thomas
Steven
26 Elhoreya Street
01234567899
sthomas@gmail.com
10      6       1995

Thomas
Steven
26 Elhoreya Street
01234567899
sthomas@gmail.com
10      6       1995

Zayed
Mohsen
25 Sorya street
01112334242342
mohsen@gmail.com
13      3       1990

Ahmed
Islam
5 Fawzy Moaz street
01142425524
islam@gmail.com
4       6       1980
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```

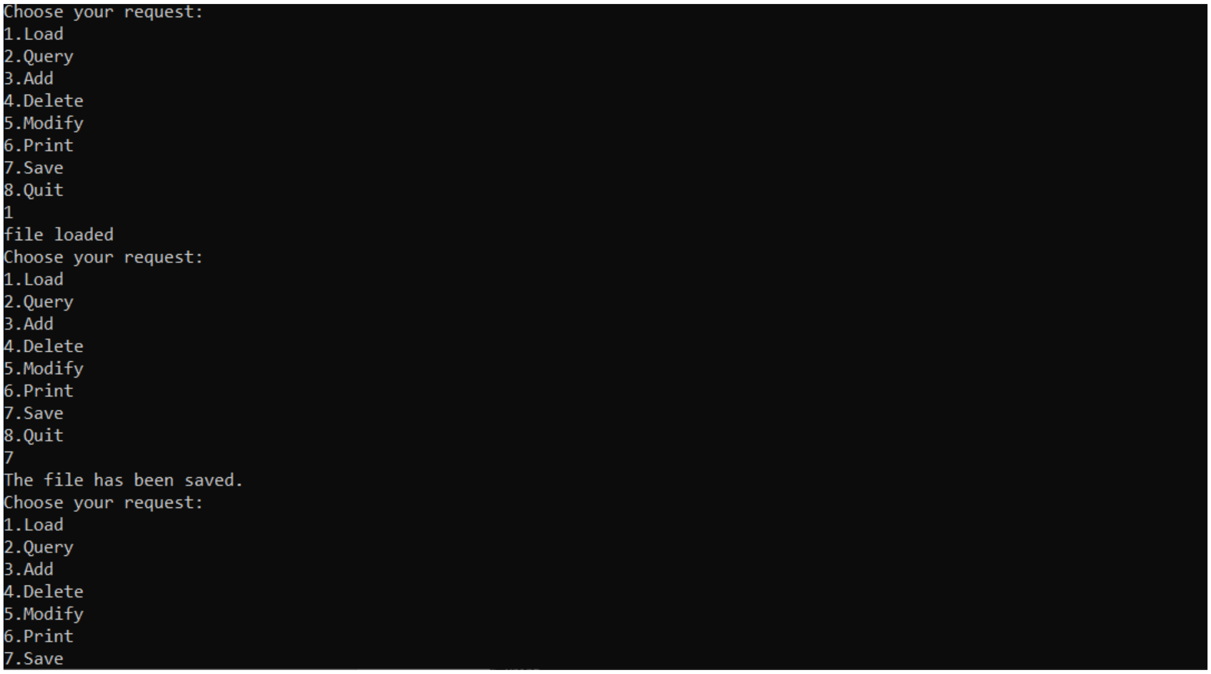
In this case the user asked to sort the file alphabetically by last name



```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
6
please select 1 to sort by name
please select 2 to sort by dob of birth
1

Islam
Ahmed
5 Fawzy Moaz street
011424255244
islam@gmail.com
```

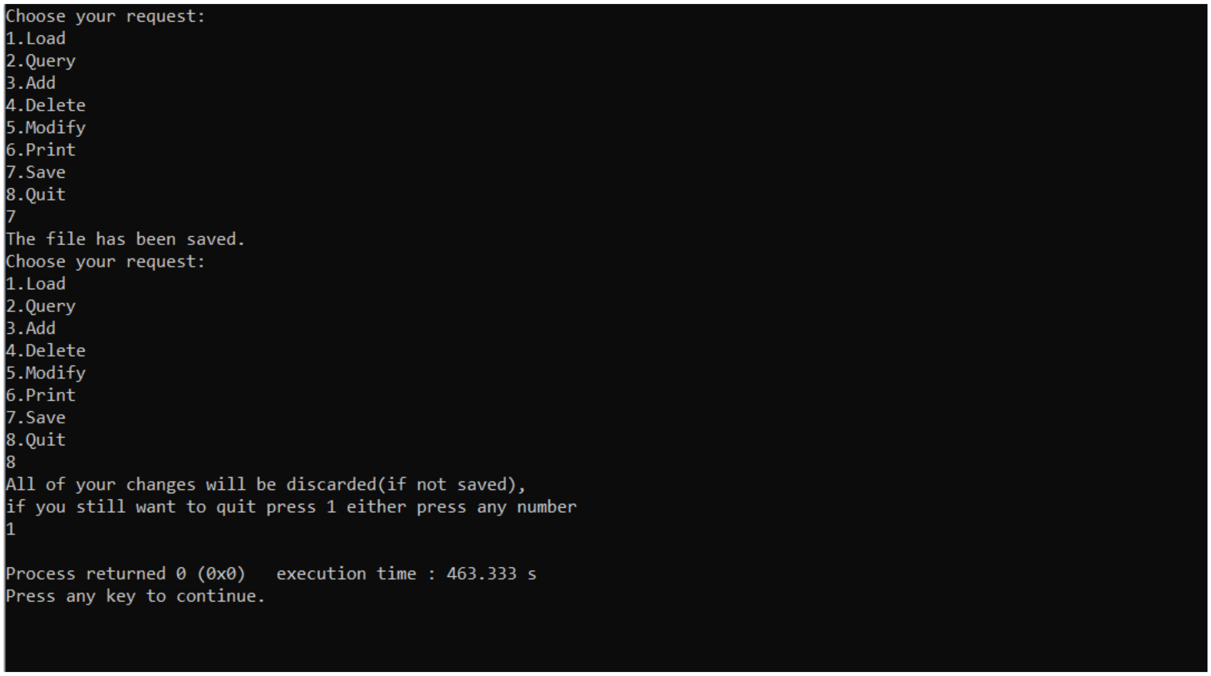
```
Ahmed
hassan
23 elhoreya street
01229993011
hassan@gmail.com
12      3       2000

Steven
Thomas
26 Elhoreya Street
01234567899
sthomas@gmail.com
10      6       1995

Steven
Thomas
26 Elhoreya Street
01234567899
sthomas@gmail.com
10      6       1995

Mohsen
Zayed
25 Sorya street
01112334242342
mohsen@gmail.com
13      3       1990
Choose your request:
```

8) Save



```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
7
The file has been saved.
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
```

9) Quit



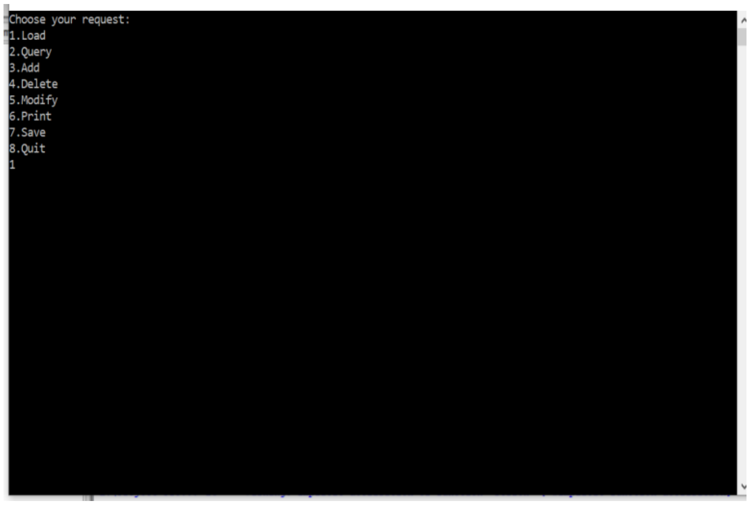
```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
7
The file has been saved.
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
8
All of your changes will be discarded(if not saved),
if you still want to quit press 1 either press any number
1

Process returned 0 (0x0)   execution time : 463.333 s
Press any key to continue.
```

3.User manual:

Load function:

In order to apply load function:

1)Enter 1 as shown in fig 3.1



```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
```

Fig 3.1

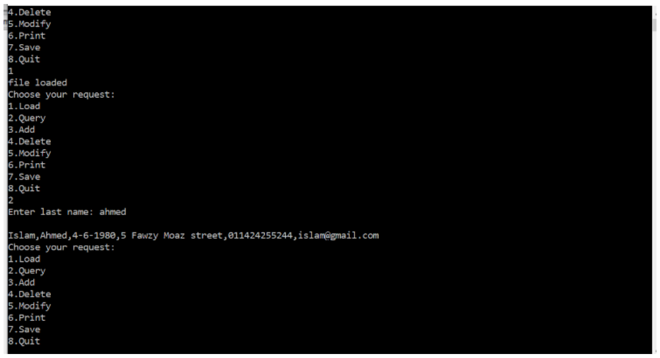
Query:

In order to apply load function: fig 3.2

1)Enter 2.

2) Enter the last name of the record you are searching for.

3) The record will be printed out

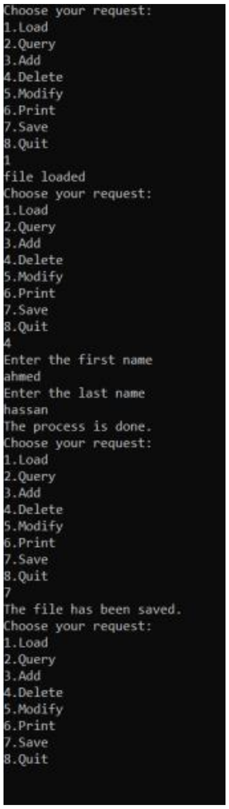


```
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
2
Enter last name: ahmed

Islam,Ahmed,4-6-1980,5 Fawzy Moaz street,011424255244,islam@gmail.com
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```

Fig 3.2

Add

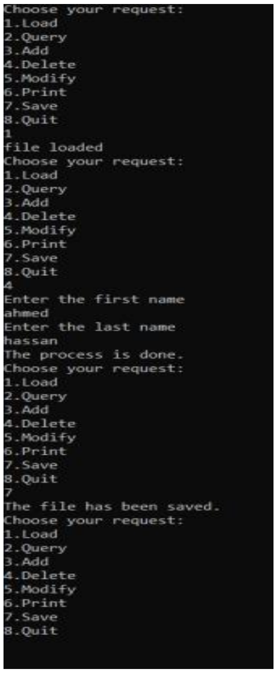
1. Enter 1 to load file.
2. Enter 3 to choose the add option.
3. Enter 1 to choose yes to confirm that you would like to add a record.
4. Enter last name of new record.
5. Enter first name of new record.
6. Enter day of birth of new record.
7. Enter month of birth of new record.
8. Enter year of new record.
9. Enter address of new record.
10. Enter phone of new record.
11. Enter email of new record.
12. Enter 7 to save new entry.



```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
4
Enter the first name
ahmed
Enter the last name
hassan
The process is done.
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
7
The file has been saved.
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```

Delete

1. Enter 1 to load file.
2. Enter 4 to choose the delete option.
3. Enter the first name that is in the record wanted to be deleted.
4. Enter the last name that is in the record wanted to be delete.
5. Click 7 to save changes.



```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
4
Enter the first name
ahmed
Enter the last name
hassan
The process is done.
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
7
The file has been saved.
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```

Modify

1. Enter 1 to load.

2. Enter 5 to choose the modify option.

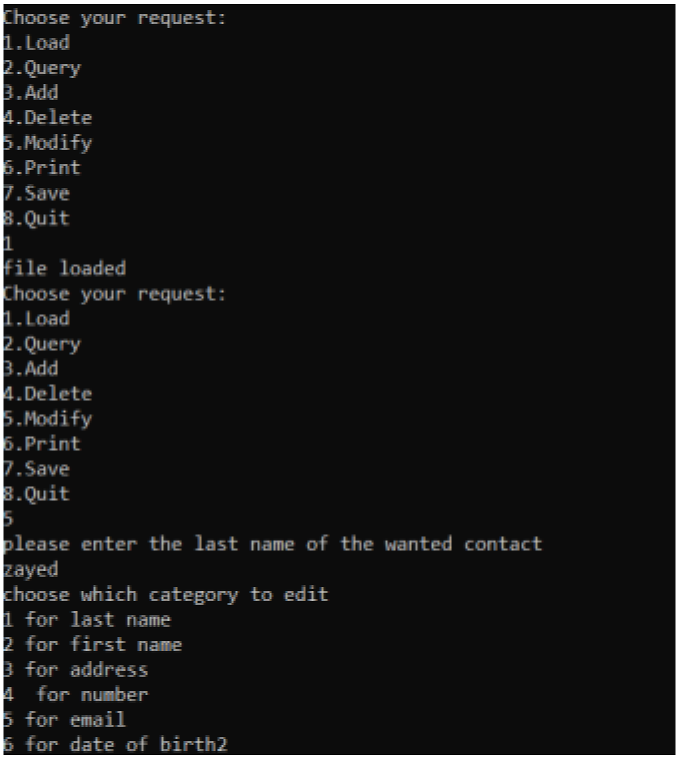
3. Enter the last name of the record you need to modify.

4. In case of duplation in last names all the users with that last name will be printed on the screen so please input the number next to the wanted contact.

5. Enter the number next to the area that needs to be modified.

6. Enter the modification.

7. Enter 7 to save.



```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
5
please enter the last name of the wanted contact
zayed
choose which category to edit
1 for last name
2 for first name
3 for address
4  for number
5 for email
6 for date of birth2
```

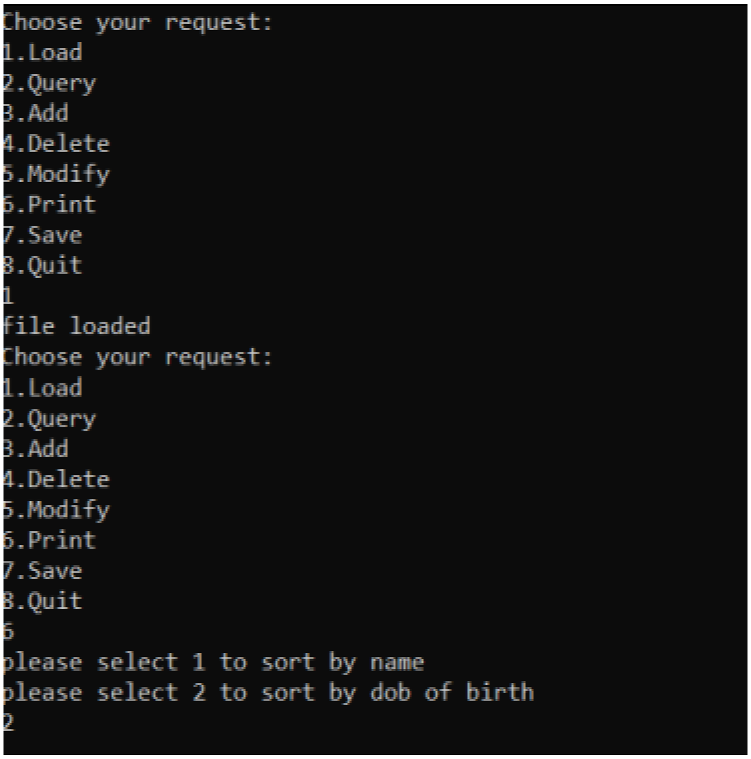
Print(sort)

1. Enter 1 to load.

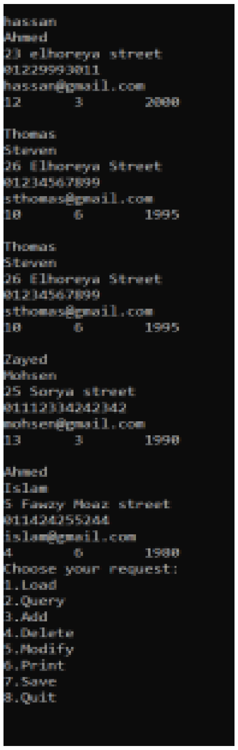
2. Enter 6 to choose the print(sort) option.

3. Enter 1 or 2 depending on the type or sorting wanted whether by date of birth or alphabetically.

4. The file will be printed on the screen in the specified method.



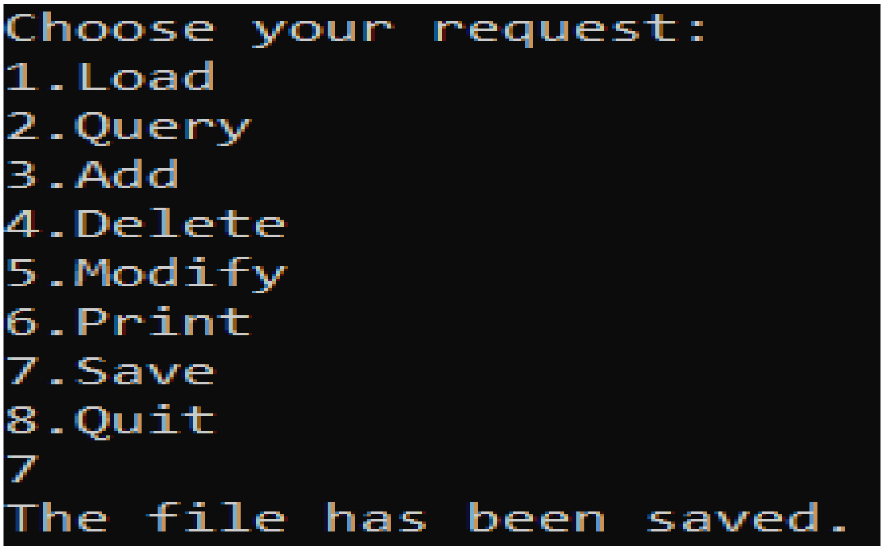
```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
1
file loaded
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
6
please select 1 to sort by name
please select 2 to sort by dob of birth
2
```



```
hassan
Ahmed
23 elhoreya street
01229993011
hassan@gmail.com
12      3       2000

Thomas
Steven
26 Elhoreya Street
01234567899
sthomas@gmail.com
10      6       1995

Thomas
Steven
26 Elhoreya Street
01234567899
sthomas@gmail.com
10      6       1995

Zayed
Mohsen
25 Sorya street
01112334242342
mohsen@gmail.com
13      3       1990

Ahmed
Islam
5 Fawzy Moaz street
01142425524
islam@gmail.com
4       6       1980
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
```

save

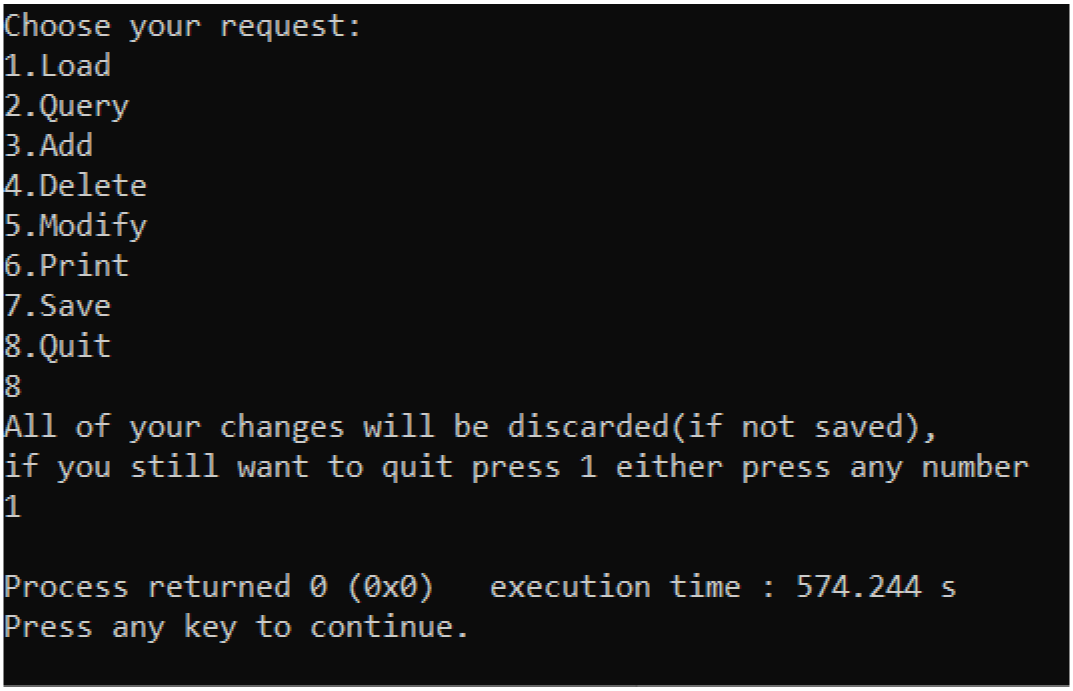
To save data after applying your commands, press 7 as shown in the figure below.



```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
7
The file has been saved.
```

Quit

In order to exit the program, press 8 as show in the figure below.



```
Choose your request:
1.Load
2.Query
3.Add
4.Delete
5.Modify
6.Print
7.Save
8.Quit
8
All of your changes will be discarded(if not saved),
if you still want to quit press 1 either press any number
1

Process returned 0 (0x0)   execution time : 574.244 s
Press any key to continue.
```

4. Algorithms:

4.1Search Algorithms:

In order to search for the required record, Linear search algorithm is used. Linear search is a simple process as it compares each element of the array with the key value for example in Query function, the key value is the last name given by the user as an input then this key value is compared to each last name in every record loaded from the file of the phone directory. Moreover, a for loop is used to search every record for the key value using variable i as an index for every row in the array until i becomes greater than or equal count, which is the number of records in the file, as shown in code 4.1. strcasecmp(,) is used to compare two strings which returns zero if they are equal.

```
for(i=0; i<count; i++)
{
if(strcasecmp(x[i].last,lastname)==0)
```

Code 4.1

The output of the search algorithm varies according to the purpose of the program, but in this program in query function the output will be printing the first name, last name, address, email and phone number for all users of that name and the flag will be equal 1. If the record isn't found the program will print "Search not found". This check is done by testing if the flag equals zero.

The linear search algorithm was also used in the modify function. The ideas behind the modify is that after the data is loaded in an array that array is searched for the matching last names if 1 was found it will ask which needs to be edited and takes the new input, otherwise if more than one last name was found an array of indices is made to contain the locations of the contacts and all of them are printed for the user to choose which one, he wants to edit.

4.2 Sorting Algorithms:

The sorting algorithm used in this program is bubble sort. Bubble sort is a simple algorithm that repeatedly swaps the adjacent elements, if they are in wrong order, by making several passes through the array. The pass through the list is repeated until the list is sorted.

The algorithm in print is to make an if function to ask which type of printing in required for the date of birth sort the contacts are sorted by bubble sorting by years first and in case of existence of common years, only the contacts with the common years will be sorted according to the months and the same applies for days if more than one contact has the same year and month of birth.

For the alphabetical sorting bubble sorting was also used but this time although the sorting is required to be according to the last name it sorts according to the first name first and then it sorts that sorted array one more time according to the last and this is to deal with the duplication of last name case so in this way the contacts with the same last name would be sorted according to the first name however if both first and last names are repeated they will be placed after each other In no specific order but in the correct position compared to the rest of the contacts.