PHONEBOOK PROJECT

Introduction:

This program enables the user to deal with large number of contacts and let him/her do the following :-

1- Add a contact to the phonebook.

2- Delete a contact from the phonebook.

3- Edit a contact in the phonebook.

4- Search for a contact in the phonebook.

5-Load a file that contains the contacts that the user wants to deal with.

6- Save the new contact in a new file after doing one or more of the previous options.

7- Print the whole directory.

===============================================================================

How To Use The Program:

The program asks the user to choose one of the options from the menu.

The user must enter a number from 1 to 8 to choose the desired operation.

All the program's options are optional.

1- The "LOAD" option enables the user to use contacts that exist in a file as it separates the file into contacts and each contact into a structure.

2- The "SEARCH" option is used to look for a certain contact having the last name that the user typed. If more than one contact have the last name that the user is searching for, all the contacts in the memory are printed.

3- The "ADD" option lets the user enter a new contact into the memory and then the number of contacts in the structure is increased by one.

4- The "DELETE" option asks the user to enter the last name of the contact he wants to remove. "DELETE" removes a contact from the structure and the contacts below this contact are shifted one place upwards.

EX: if a contact's last name "MOHAMED" is found in the 4th position in the array of structures and a contact's last name "AHMED" is found in the 5th position in the array of structures and "Mohamed" was deleted, The contact "MOHAMED" is removed and the contact "AHMED" becomes in the 4th position.

If more than one contact was found having the same last name the program asks the user to choose the number of the contact he wants to delete.

5- The "Modify" options enables the user to edit a certain contact. The user is asked if he wants to edit each field individually. If the user enters 'y' when asked, the program asks the user to write the new entry in this field. If the user enters 'n' the program asks the user if he wants to edit the next field.

6- The "PRINT DIRECTORY" option prints all the contacts which exist in the memory sorted alphabetically by last name and if more than a contact have same last names this option sorts the contacts alphabetically by first name.

7- The "Save" option stores the data after being processed into a new file. This option asks the user to enter the path where he wants to create this new file. If the user enter a path without folder EX: NewFile.txt the program creates a new file inside the folder of the program by default.

8- The "EXIT" option quits the program without saving the processed data into the new external file.

9-If The user enters any other number a message appears to warn the user that he entered an invalid choice.

Algorithms:

1- Sorting : the "SORT" function Is done by comparing two contacts by last name. If a contact's last name starts with a letter before the starting letter of the other contact alphabetically , the first structure will be copied into a third temporary structure and then the second structure will be copied into the first structure and finally the temporary structure will be copied into the second structure and the structures will be sorted alphabetically.

CODE:

for(i=0; i<numberofcontacts; i++)

for(j=0; j<numberofcontacts-1; j++)

{

compare1=strcmp(stock[j].lastname,stock[j+1].lastname);

if(compare1>0)

{

strcpy(t[0],stock[j+1].lastname);

strcpy(stock[j+1].lastname,stock[j].lastname);

strcpy(stock[j].lastname,t[0]);

strcpy(t[0],stock[j+1].firstname);

strcpy(stock[j+1].firstname,stock[j].firstname);

strcpy(stock[j].firstname,t[0]);

strcpy(t[0],stock[j+1].phone);

strcpy(stock[j+1].phone,stock[j].phone);

strcpy(stock[j].phone,t[0]);

strcpy(t[0],stock[j+1].adress);

strcpy(stock[j+1].adress,stock[j].adress);

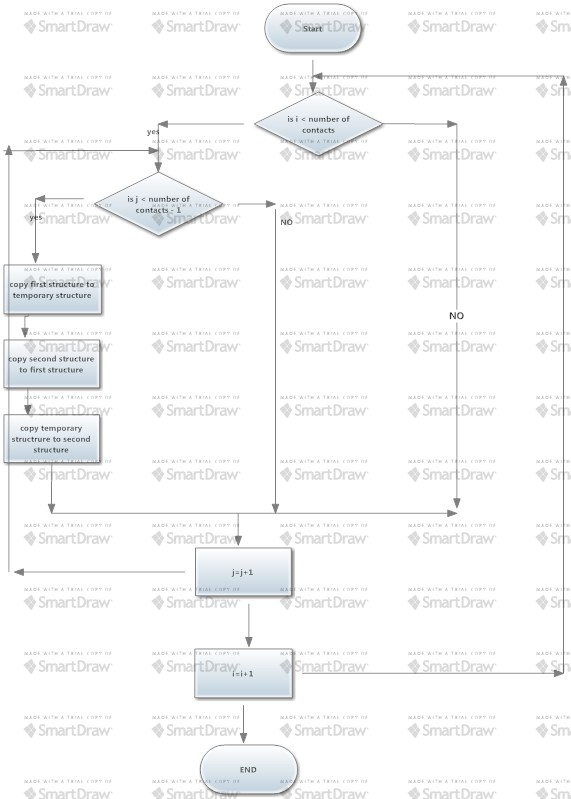
strcpy(stock[j].adress,t[0]);

strcpy(t[0],stock[j+1].city);

strcpy(stock[j+1].city,stock[j].city);

strcpy(stock[j].city,t[0]);

}



If two contacts were found having the same last name this function will compare between the first names by using the same alghorithm as following :-

CODE:

if(compare1==0)

{

compare2=strcmp(stock[j].firstname,stock[j+1].firstname);

if(compare2>0)

{

strcpy(t[0],stock[j+1].firstname);

strcpy(stock[j+1].firstname,stock[j].firstname);

strcpy(stock[j].firstname,t[0]);

strcpy(t[0],stock[j+1].lastname);

strcpy(stock[j+1].lastname,stock[j].lastname);

strcpy(stock[j].lastname,t[0]);

strcpy(t[0],stock[j+1].phone);

strcpy(stock[j+1].phone,stock[j].phone);

strcpy(stock[j].phone,t[0]);

strcpy(t[0],stock[j+1].adress);

strcpy(stock[j+1].adress,stock[j].adress);

strcpy(stock[j].adress,t[0]);

strcpy(t[0],stock[j+1].city);

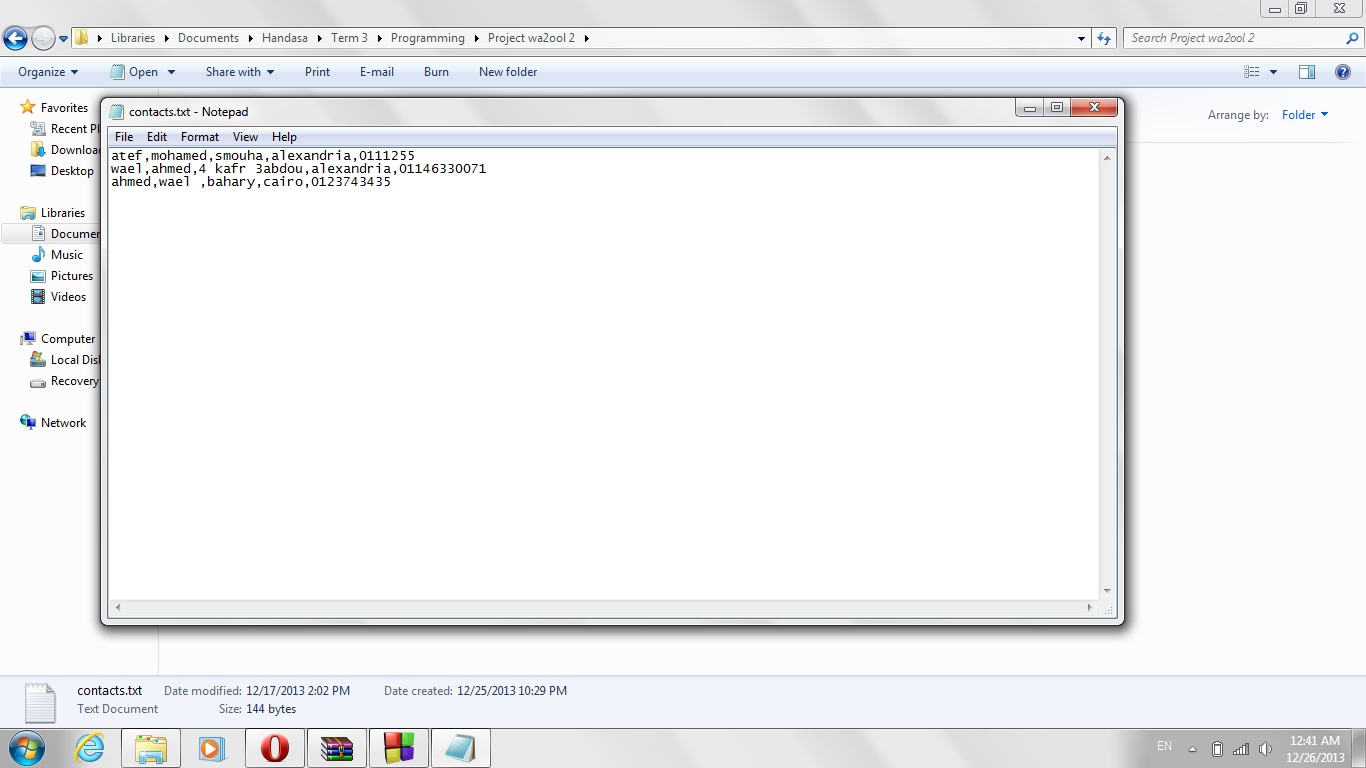
strcpy(stock[j+1].city,stock[j].city);

strcpy(stock[j].city,t[0]);

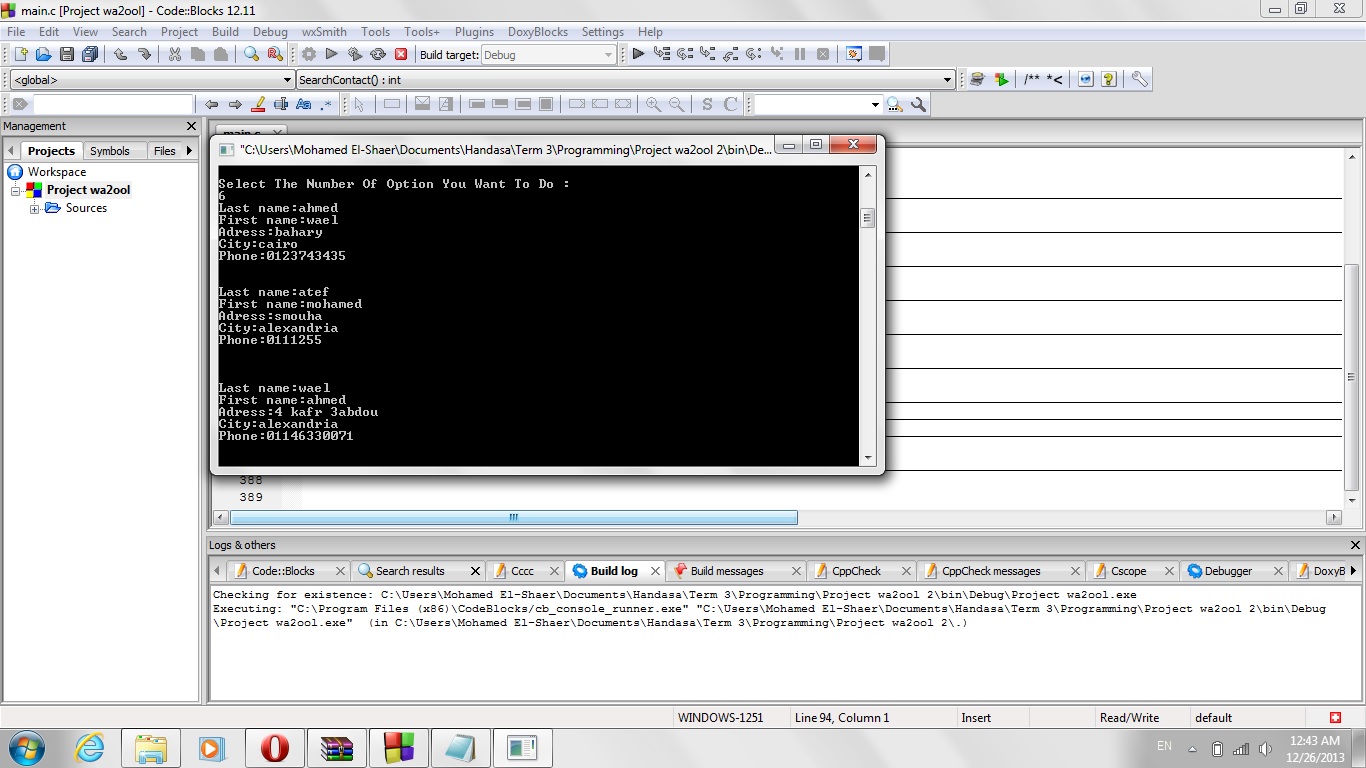
}

}

}



This figure shows the original file.



This photo shows the printed contacts after being sorted.

2- Searching : This function compares the entered last name with every contact in the structure and the result is all the contacts that have this last name.

CODE:

int SearchContact ()

{

int z;

int m;

int sub;

int flags=0;

printf("Enter The Last Name Of The Contact You Want To Look For : ");

getchar();

gets(last\_name);

for (i=0; i<numberofcontacts; i++)

{

z=strcmpi(last\_name,stock[i].lastname);

if (z==0)

{

printf("\n%d--\nlastname :%s\n",i,stock[i].lastname);

printf("firstname:%s\n",stock[i].firstname);

printf("adress :%s\n",stock[i].adress);

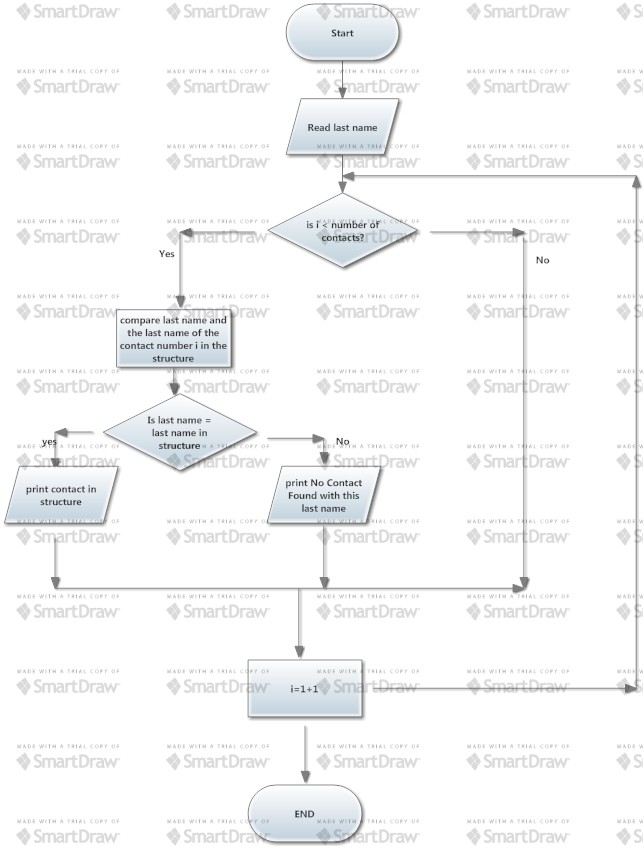
printf("city :%s\n",stock[i].city);

printf("phone :%s\n",stock[i].phone);

flag=1;

}

}



Eventually:

On the behalf of all students of Computer & Communication we'd like to thank all of you :

Eng. Samy Mamdouh

Eng. Peter Abo Alim

Eng. Mai Hassan

Eng. Mohamed Mahmoud

It was a pleasure to deal with you and we hope to see you in the next terms. We wish you all the best of luck.

Ahmed Wael Said (Load-Search-Add-Delete-Print-Edit-Separation) ID:2091

Mohamed Mahmoud Atef El-Shaer (Search-Delete-Sort-Print-Flowchart-Report) ID:2211