# 2016 HKDSE

# CIT School-Based Assessment (SBA) report

Information and communication technology (SBA report)

Option D: Software development

Title: Phone Book Directory

# Content

# 1. Objective

## 1.1 Problem & Situation

Nowadays, people put emphasis on science and technology. In order to do the things in a more efficient way, people depend and rely on the computer world, and it lead to an increasing trend on people who use mobile phone, computer. Also, the number of function of the mobile phone increased a lot, it implies that mobile phone become a part of our life. Phone book directory function is also an important function on the mobile phone, computer. As it makes our life become more convenience and efficient, for example, people do not memorize the phone number of friends, workmates and relatives and the phone book directory help us to keep the data temporally.

However, the problems come when they lost their technological product, they may lost all the numbers in the phone book directory. Also, most of the phone book directory do not support search function and do not even have a sort function. Although we remember some letter of the phone numbers, we do not know who he/she is or the phone number correct or not.

That' s why there are still some loopholes on the phone book directory nowadays.

## 1.2 Solving the problem

To solve this problem, the best solution is to make a phone book directory program that run on the computers. The user may use it to store the entries in the computer. If they lost the phone numbers, email addresses, they can have a backup on the computer. Beside they can also use the program to store some name, email address and cell phone numbers of some people that not usually call them and it is useless to store them in the mobile phone. Beside the problem above, the problem of sort and search in the cell phone, phone book directory, so I make these functions (Sort & Search) in my program.

Thus, it can directly solve the problem.

## 1.3 Intended user and their expectation (About the program)

The intended users of this program are basically the computer and the smart phone users.

The intended users would be expecting the program will be able to return some useful data about the phone number of the people they want to find, they can also search for the email of these people. The program can help them to reduce the time on finding some information. It can also help them to reduce and even avoid the risk of finding the wrong person data. They can change the data of the people or to delete the data of the people. The intended users would also expect the program to be user-friendly as not most of them are generally good at computer. The program they expected should show clear

instructions and directions as how to operate it, so they can use the program easily.

| Features | Function of the feature | How to meet users expectation |
|---|---|---|
| 1.Display Data | Show all the data of the people in the phone book. | They can see the information of all the people easily. |
| 2.Search | Search the data of the people by entering the name of the people. | This returns the information of what the users search. |
| 3.Add Data | Insert the new data to the phone book. (name, email, phone number) | This allows users to enter the information of the people to the phone book easily. The phone book will accumulate and save the data entered. |
| 4.Change Data (phone number) | Change one’ s phone number by entering the name of that person. | This allows users to change the phone number of the people and save it to the phone book. |
| 5.Change Data (email address) | Similar to the above function but is to change the email address of a person. | This allow users to change the email address of the people and save it to the phone book |
| 6.Delete Data | To delete the data, including the phone number, email address, name of the person by entering the name of that person. | This allows users to delete the data that is not used, and this data may permanently remove from the phone book. |
| 7.Sorting | Sort the information | Sort the information and let them see the data more clearly and find the data more easily. |

# 2. Design and implementation

## 2.1 Brainstorm

At first, I thought about what to include and design my phone book. In order to make it become more complete and have more function, so I took my mobile phone and have a reference. After having a reference from the mobile phone, Name, phone number, email address…

Finally, I choose of these items to be a part of my product.

Allow user to add contactors

Allow users to delete contactors
* Users need to delete it item-by-item

Allow users to change data:
*change the email of the contactors
*change the phone number of the contactors

Allow users to sort the contactors data by name (A-Z)

Allow users to search the contactors

Allow users to see all the data and information of all contactors

Allow users to quit the phonebook
* All the information will save when the quit option is chosen

Give a Graphical Interface to the user

Refresh the screen

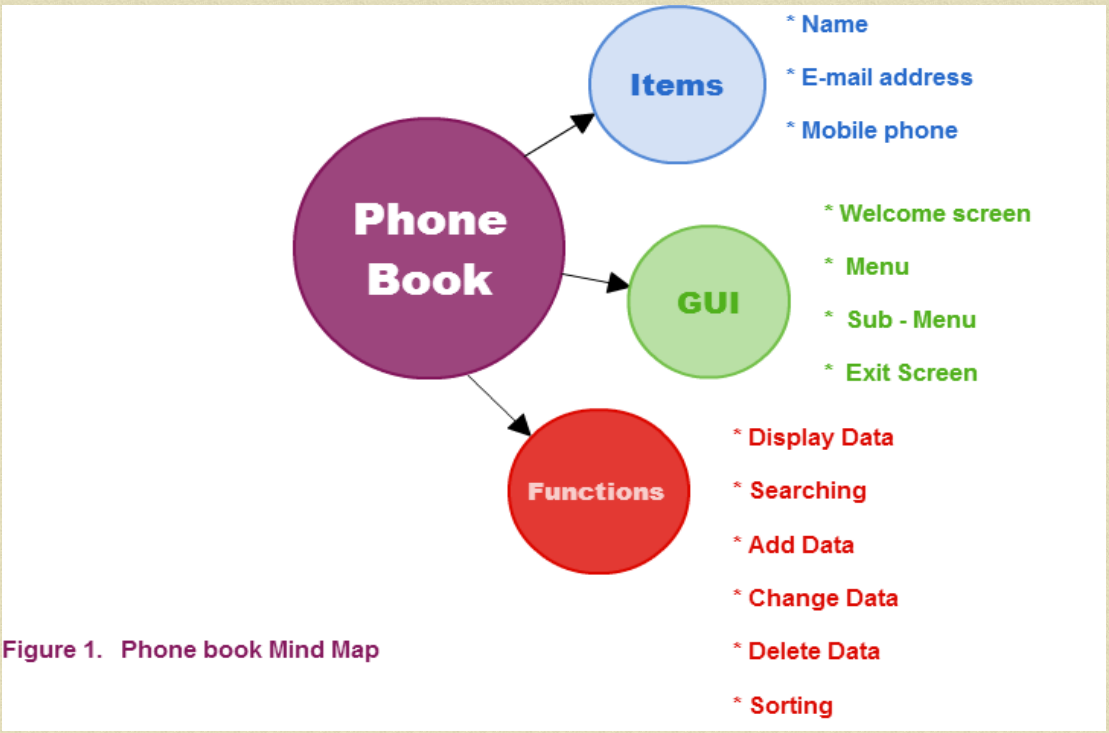



Figure 1. Phone book Mind Map

## 2.2 The users interface

When I brainstorm and wrote my program, I tend to use both of the Command Line Interface (CLI) and the Graphical User interface (GUI) . There are several reasons that why I chose to use the both of these interface.

For the command line interface, I think CLI is quite fast, the user just need to remember the commands, and they can enter the command in a more efficient way. It can help them to save a lot of time. For an example, enter the data in command line.

For the graphical user interface, I think GUI can be more user-friendly, as the picture can help the user to be more easily to use the phone book. I use the GUI for the welcome screen, the menu, sub-menu and the exit screen as well. Thus it can be more user-friendly to the user.

## 2.3 Modularization

For my program, I divided it into four sub-programs. They are insert data, change data, delete data and sorting respectively. In this program, the searching procedure is the main program of this phone book and it is the important procedure of the whole program. The reasons are that the sub-program are operated under the searching procedure, especially for the sub-program change data and the delete data, as these two sub-program are run after the searching procedure is completed. That’ s why the searching function acts as the main character of this program. Thus, some sub-program can be run after the searching.

## 2.4 The Data structure

Before I wrote the program, I have created a text file 'Phone.txt' to store the data that is needed for the program, included the name of the person, the phone number and the e-mail address. Also, I have used three arrays to store the name, phone and email respectively. The reasons for me to make a text file that it can help me to be more easily to manage the data, just like the text file 'Phone.txt', I can read the data of the text file from the program and write something into the text file by using the Readfile and Writefile in my program. That's why I tend to use the text file to store the data instead of writing the record and the information to my program. It can help to make my program become clearer, simple and easily to control, edit the data. It is also beneficial for me to have the data validation and the debugging works. Thus, I prefer to make a text file to store the data.

For the array, I think the reason for me to use the array is that the array can help to store the data in a more efficient way. As array are used to group similar data together and to make it easier to search and sort. Due to this reason, I guess array is quite suitable for my program. As my program has three main data information which are the name, phone and email address, using the array can help to manage the program and the source code more easily. The other reason is that using the array can help to reduce the redundant of the list of data, record, for an example, make a phone book of say, 100 people, rather than make a bunch of variables named 'person1, person2, person3....person100', using the array to avoid this problem and can make the program more clearer. Also, since the data is stored in the array, the program can access the data faster and directly. The program can be operated more efficient as the redundant of the program coding not appear. Therefore, I tend to use the array and the text file to store the data.

## 2.5 Constant and Limitations

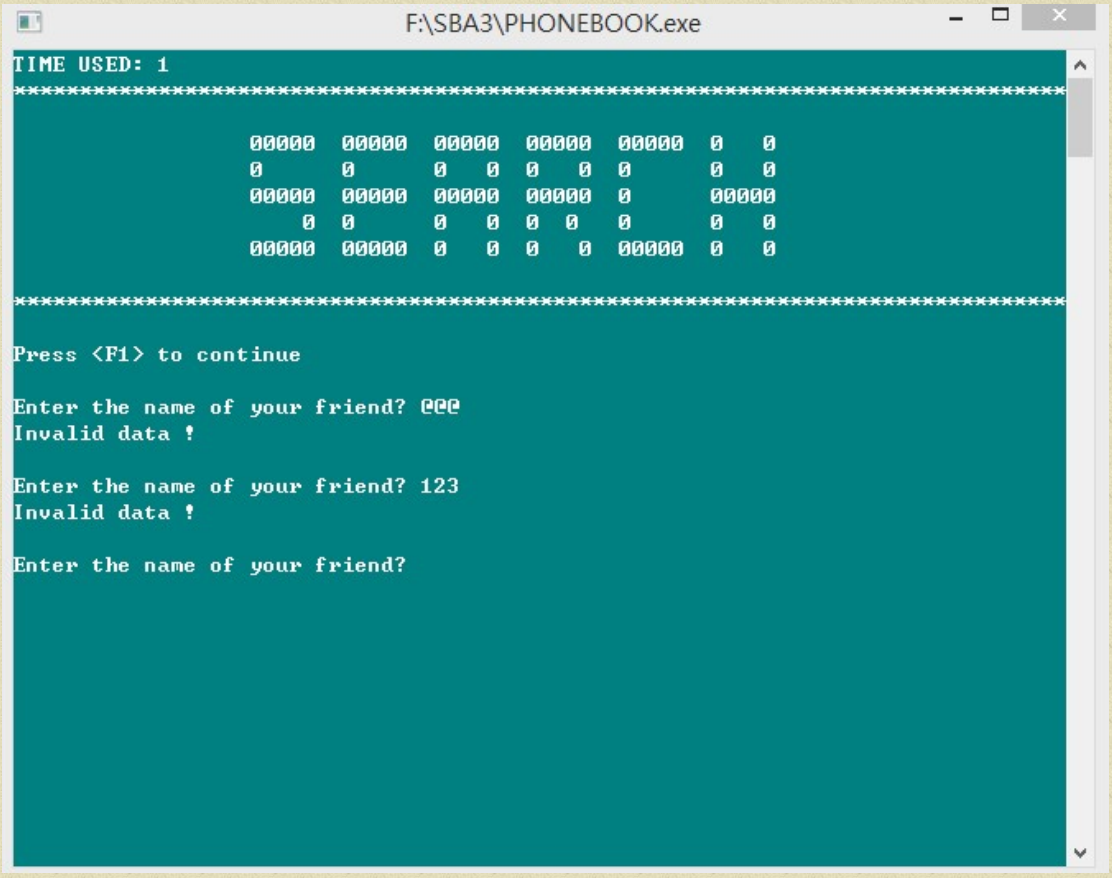
For this program, I set a constant Max which represent the maximum number is 100. Due to I set the constant of the Max is 100, the array of the name, phone, and the email address are both 100 {[1…Max] of String}. The reason for me to set the constant amount become 100 is that I think the scale 1 to 100 is enough for a phone book directory. It is because the number of 100 is large enough to store 100 person data. Also, the maximum data can be changed and reset easily, the user just need to change to constant number of Max. When consider the demand of people (program user) and their job, the scale of the record size and the amount of the constant number can be changed into a larger amount of even a smaller amount according to the need of the user, and it is flexible for user. So that, I chose to set the number of Constant become 100 since I think the amount of 100 is a reasonable number.

## 2.6 Data Control

In order to prevent and reduce the input error, it is necessary for doing some validation check after entering the data into different sub-program, as it can help to check whether

there are any typing mistakes or typing error. Also, it is hard to find out and prevent the typing error, so via the validation check; it can as much as possible to reduce the input error.

First of all, for the validation check, there is a validation check in each sub-program. In the searching function, a user need to enter the name of a person to check the data of that person, but if the user enter the name neither a large capital nor a small capital, the program will only return a sentence ‘ invalid data ‘. The searching function may judge whether there are any typing errors, for example, if some symbol or number is added, the program may return the sentence ‘invalid data’ until the valid data is entered.

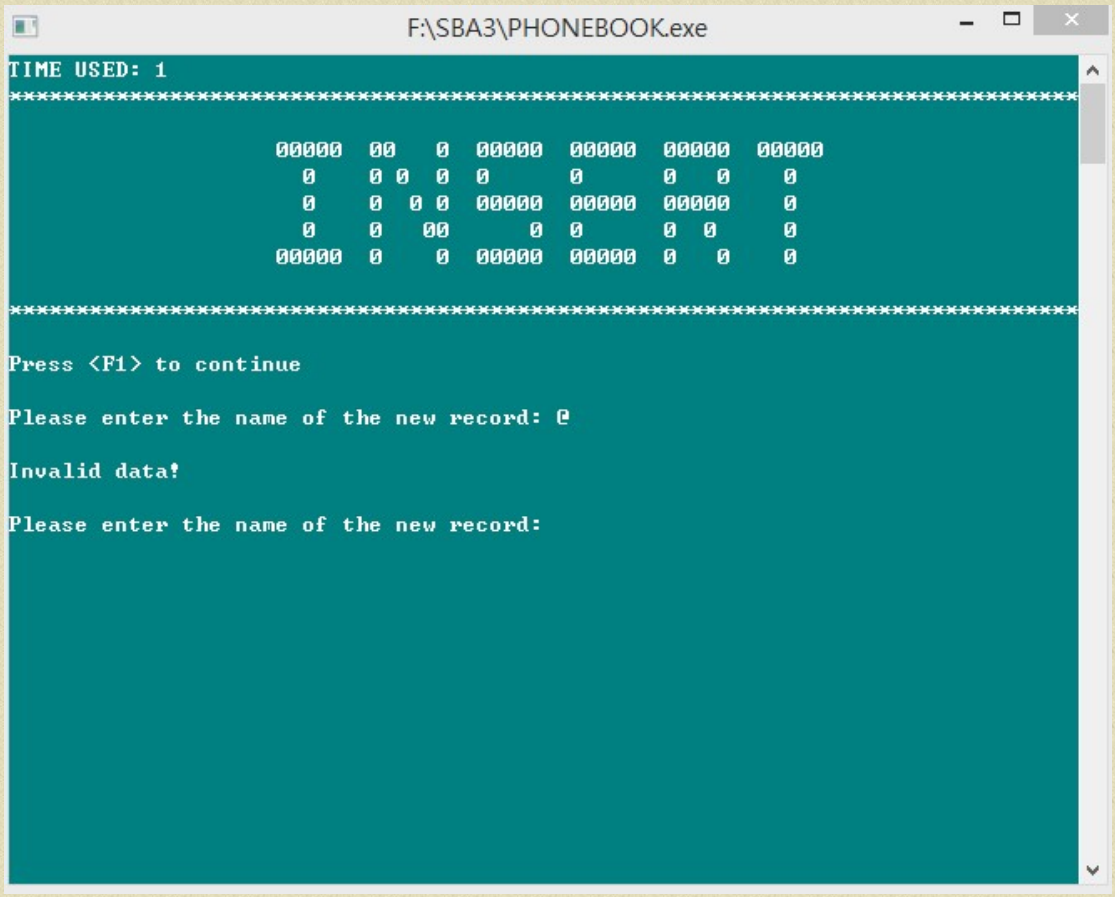



After the input passes the checking, the procedure searching will start and to find out whether the name entered exist or not. Thus, this checking method can help to prevent the typing error of the users as much as possible.

Besides of that, the other sub-program insert data also have a validation check. There are three types of data that the users have to enter; they are name, phone number and the email address. There are both validation checks in the data of name and phone number.

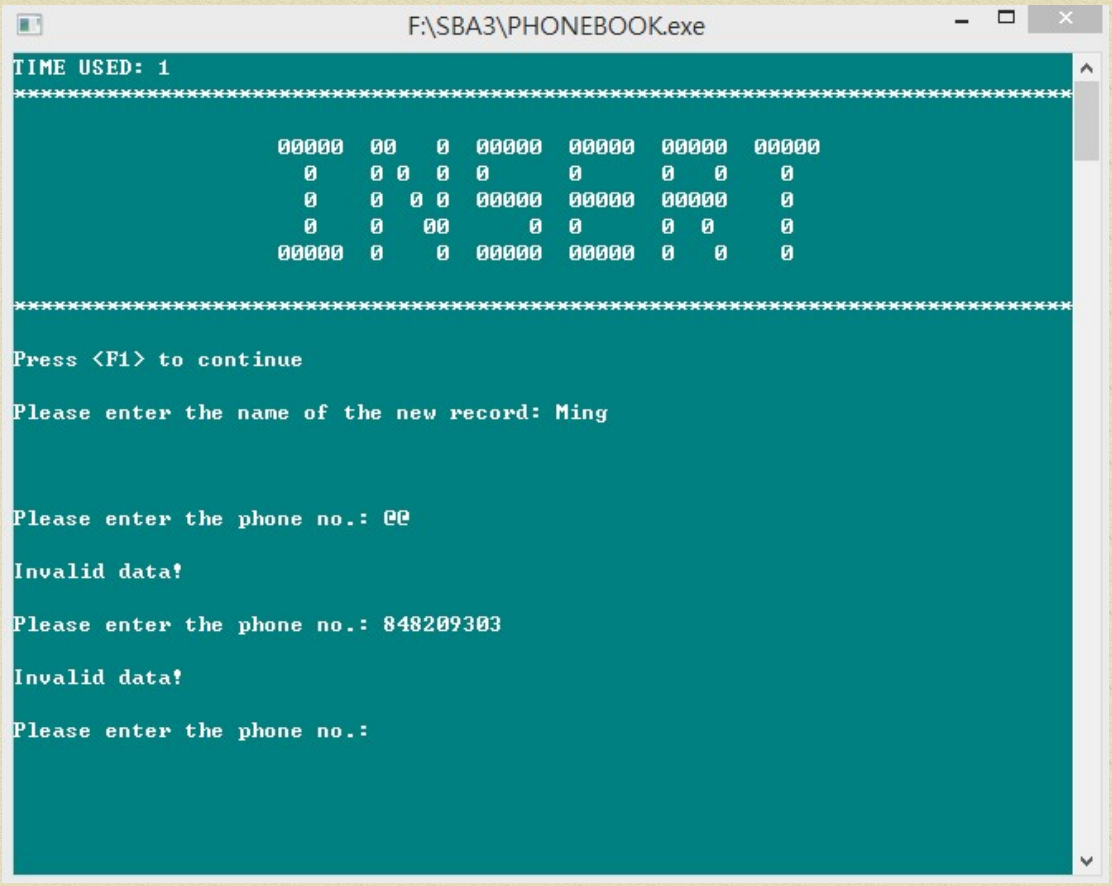
For the name, the user must be type in small capital or large capital, if a user enter in neither a large capital nor a small capital, the program will return the sentence ‘invalid data’, and the loop will until the user enter the right information. Also, if the user enters

the name of a person with more than 25 words, the program will also return the sentence ‘invalid data’. As the people are easily to enter the repeat word or the symbol, so after using the validation check, it can help to reduce the user to enter the incorrect words and it can also help to reduce the time of searching the data in a incorrect format. Therefore, it is indeed for the validation check to filter the wrong and unsuitable format of data.

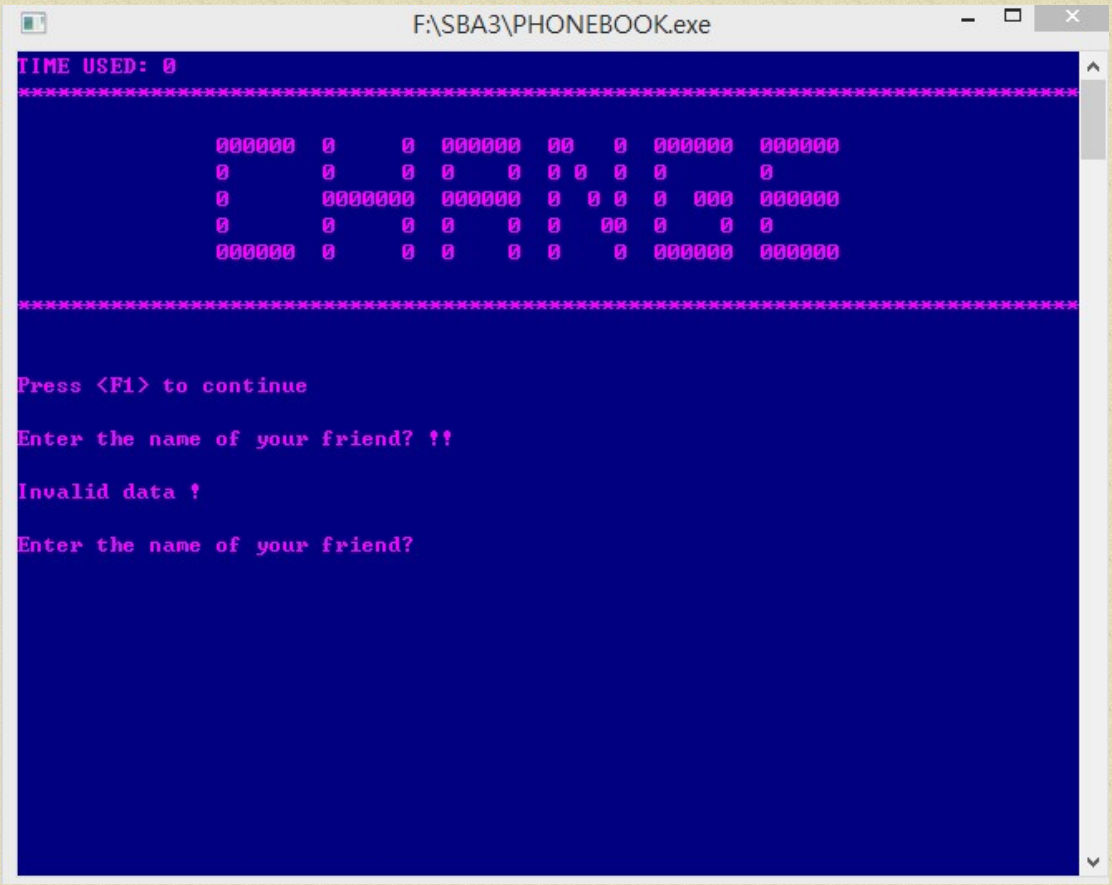



For the phone number inserting, the user can only enter the number 0 to 9. Also, the user can only enter in 8 digit phone numbers, no any 9 digit phone numbers or 7 digit phone numbers, otherwise the program may also return the sentence ‘invalid data’ until the right integer is added. After the right format of data is added, the procedure of the program will continue to go on. Thus, the validation check can take responsibility of the

input data.




For the change data and delete data, they are also using the same validation check to check whether the format of the name and the phone number is correct and to prevent the loopholes of the program. Also, the user can be easily to confirm they have entered the right data into the program. Thus, the validation check is suitable for the phone book.





These validation check will mainly use in the searching procedure, insert procedure and the change procedure. Under the validation check, lots of the mistake can be reduced.

## 2.7 The Algorithms applied

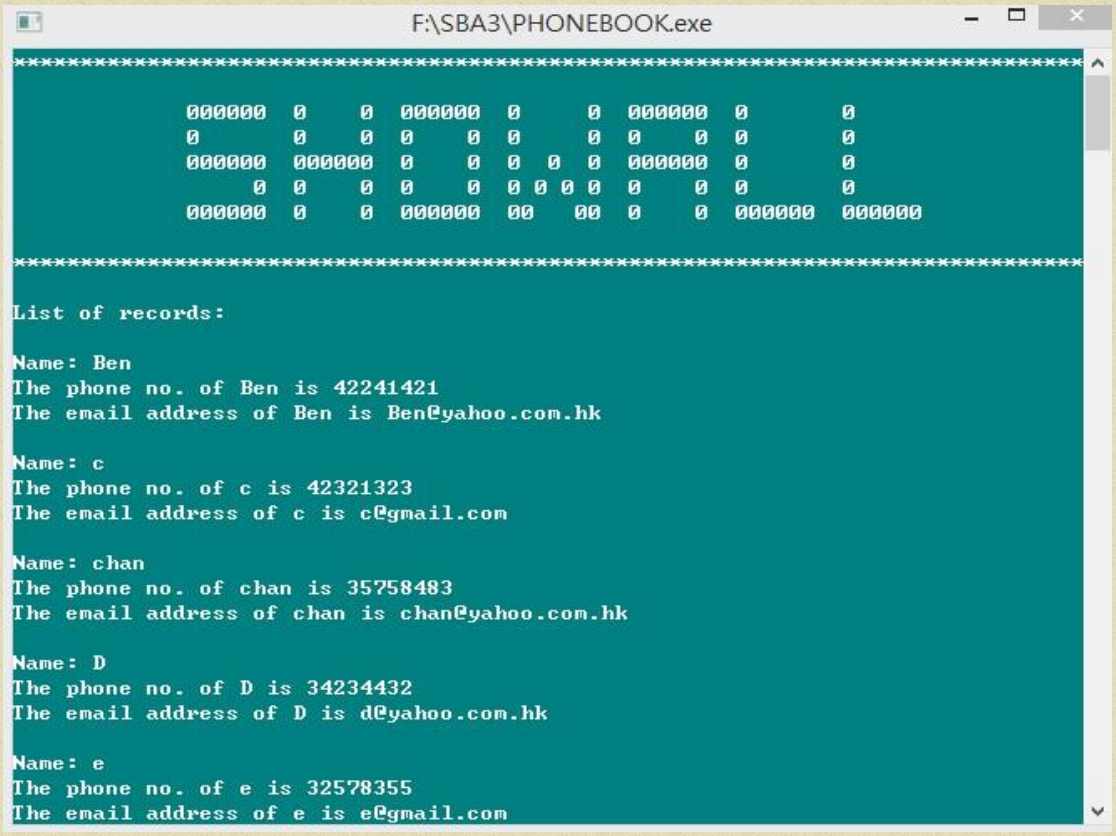
In this program, I tend to use the sorting algorithms and I use a bubble sort. The reason for me to use a sorting algorithm is that a phone book needs to store a group of people. In order to make it more user-friendly, to allow user more easily to find out the

data and information of the people, using a bubble sort can use to sort the people by name, follow by the character A to Z. The users just need to enter the function of sorting, and the process of sorting will run automatically. It is easy for user to sort the data. After the process of sorting, all the data will be clearer, and the data after sorting will be shown on the function Display Data. Therefore, I believe the searching function can be beneficial for the users and allow them to find the target person in a more efficient way.

Also, I have used the linear searching in my program. The reason for me to use the linear search is that the linear search is faster than the binary search. The other reason is that the scale of the data is not really large, so using a linear search is more suitable for the program. That’ s why I chose to use the linear searching.

## 2.8 User guide of the program

1.

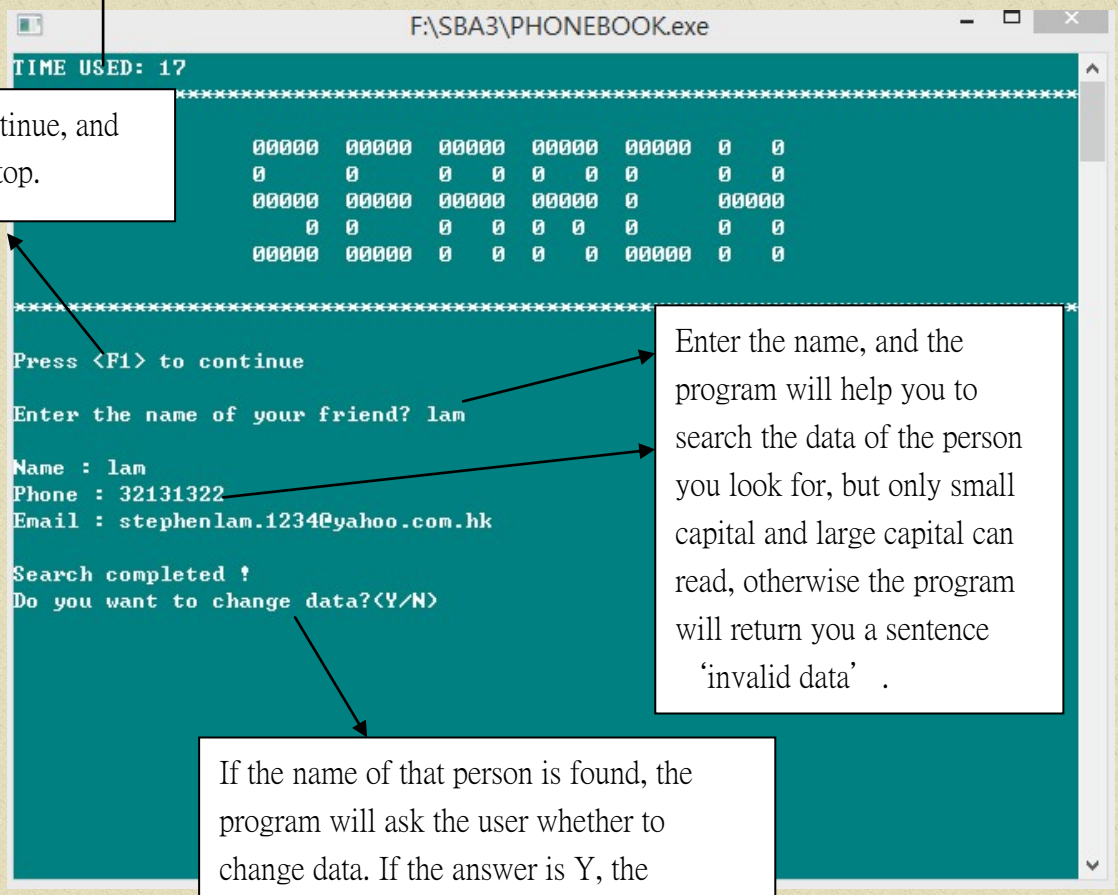



Enter the option 1, and the users can see all the data from the phone book.

## 2. Searching

To count the time used

Click F1 to continue, and the timer will stop.



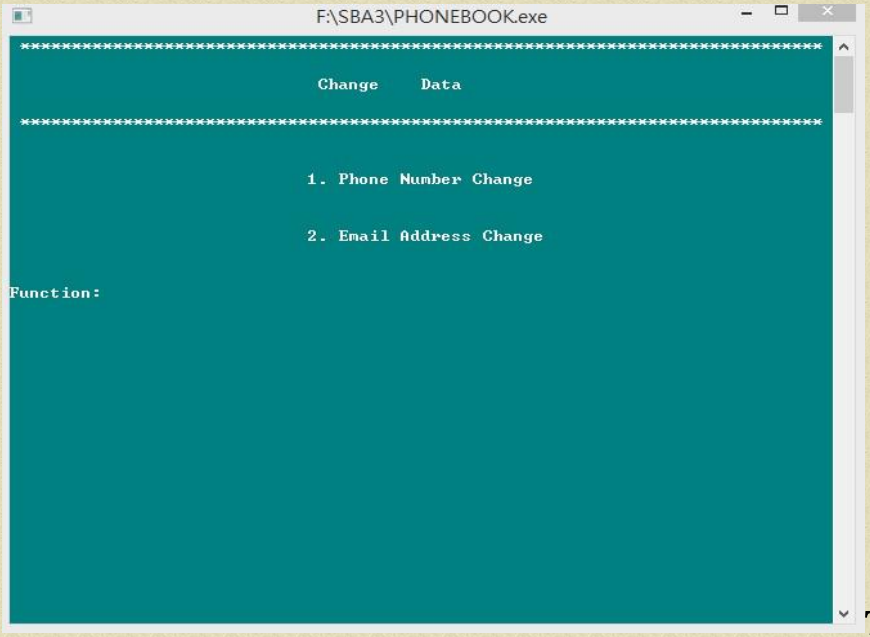

Enter the name, and the program will help you to search the data of the person you look for, but only small capital and large capital can read, otherwise the program will return you a sentence 'invalid data'.

If the name of that person is found, the program will ask the user whether to change data. If the answer is Y, the program will move to the change data function.

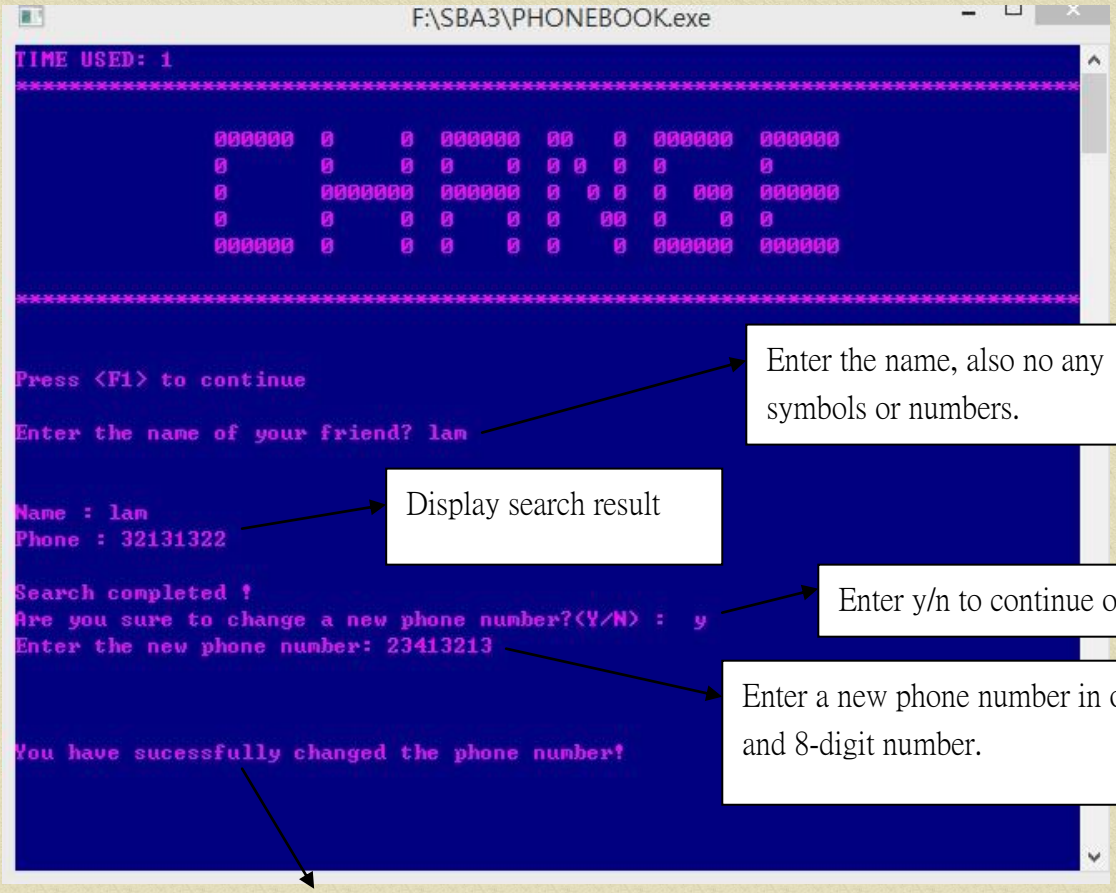
## 3. Add data


After doing the above steps, the data can be written into your phone book and also the text file.

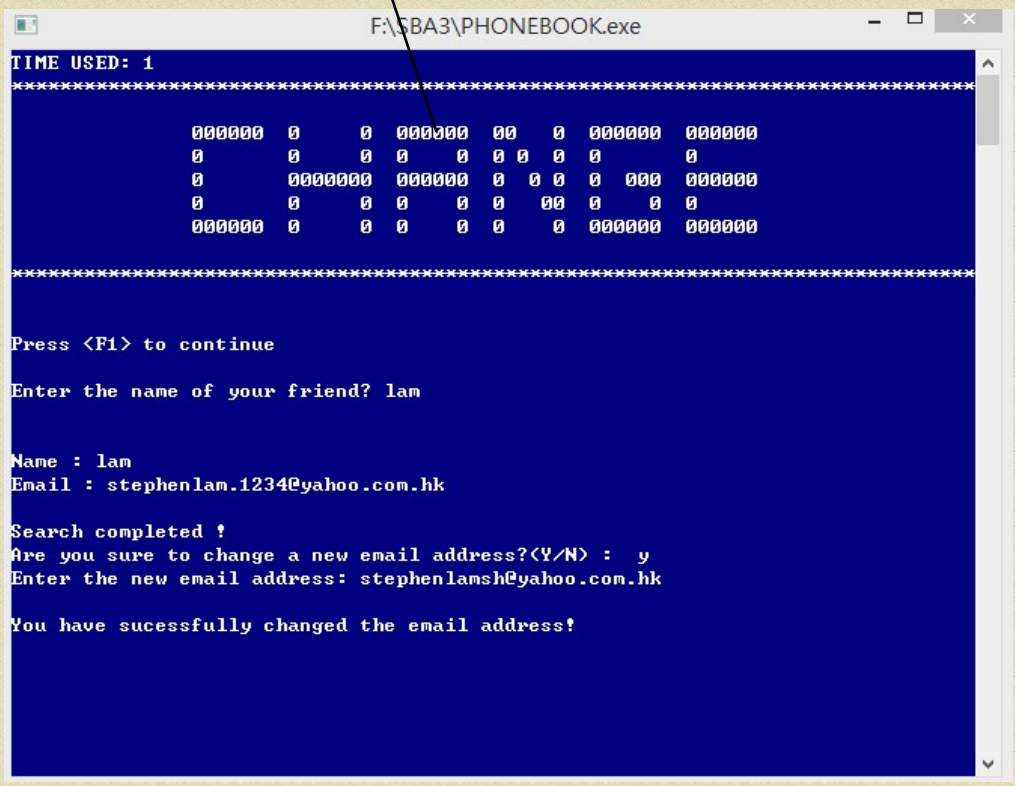
# 5. Change Data




There is a sub-program for you to choose to change the phone number or change an e-mail address.



Similar to the above change phone number procedure.



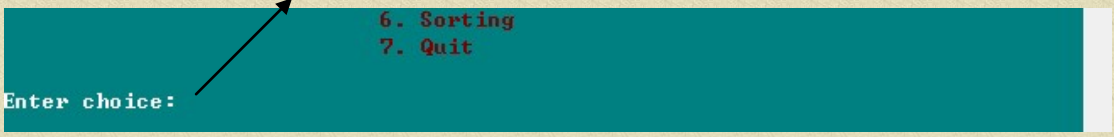


Enter the name of the person in only the small capital and large capital.

Enter y/n to continue. If answer is y, then the record will be deleted. If the answer is n, the program will return to the menu page.

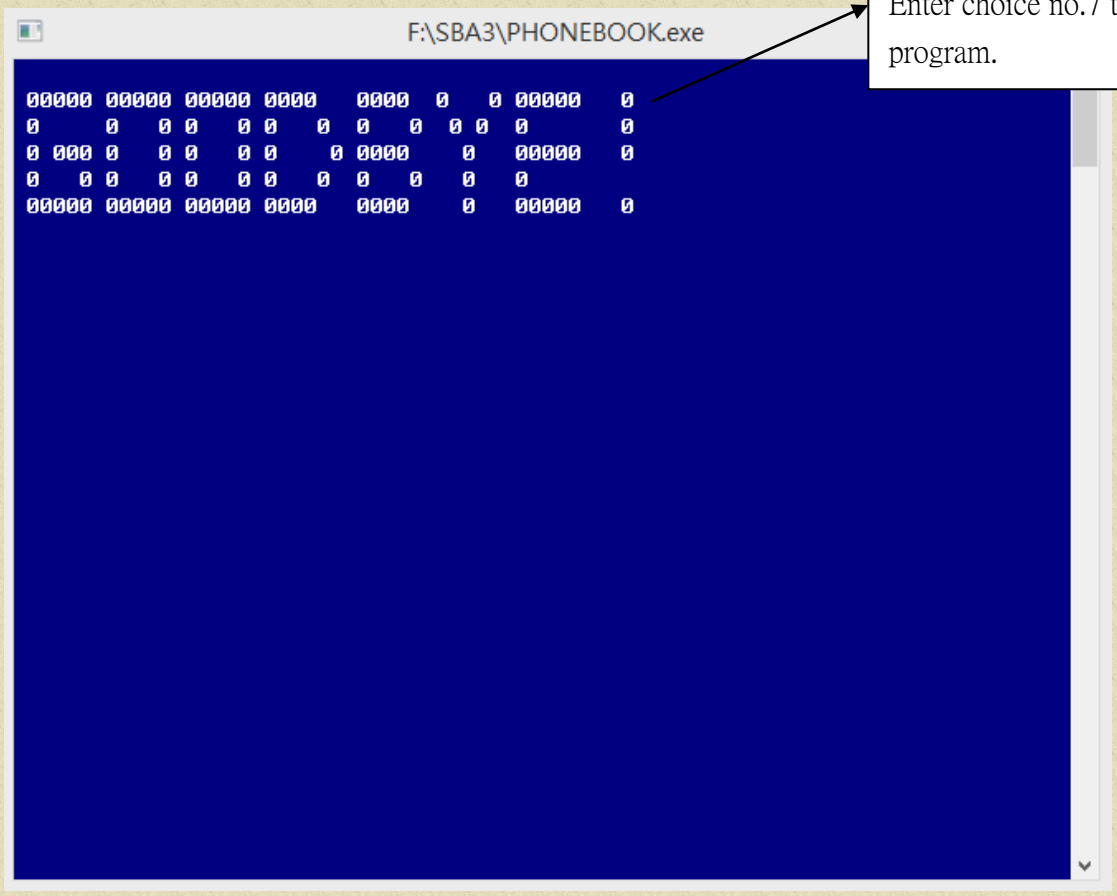
## 6. Sorting

The users' just need to enter choice no.6 and the program will sort the name of the person according to A to Z automatically.




## 7. Quit

Enter choice no.7 to exit the program.

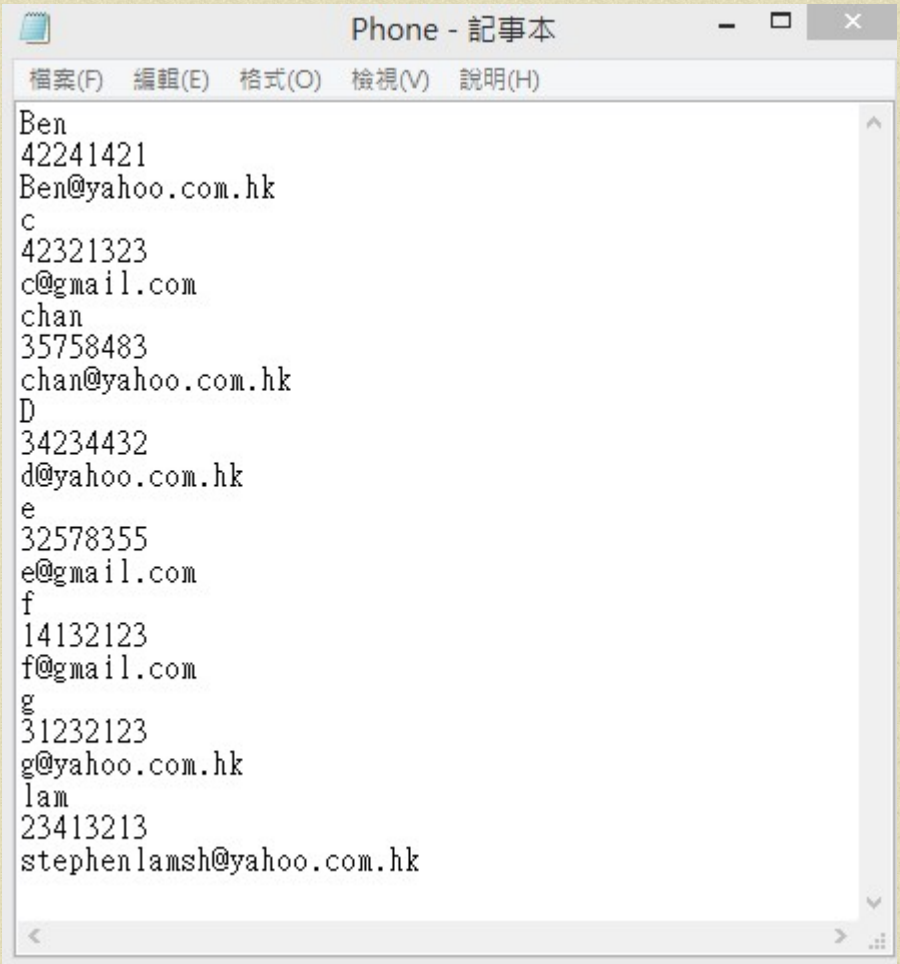


# 3. Testing and evaluation

## 3.1 Testing

I will test my program using the notepad. I will first save a group of data; they are name, phone number and email address respectively. I will check the function of my program can successfully run!

Notepad- 'Phone.txt'

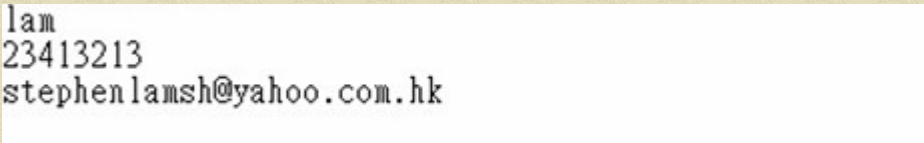


Test 1 Display Data

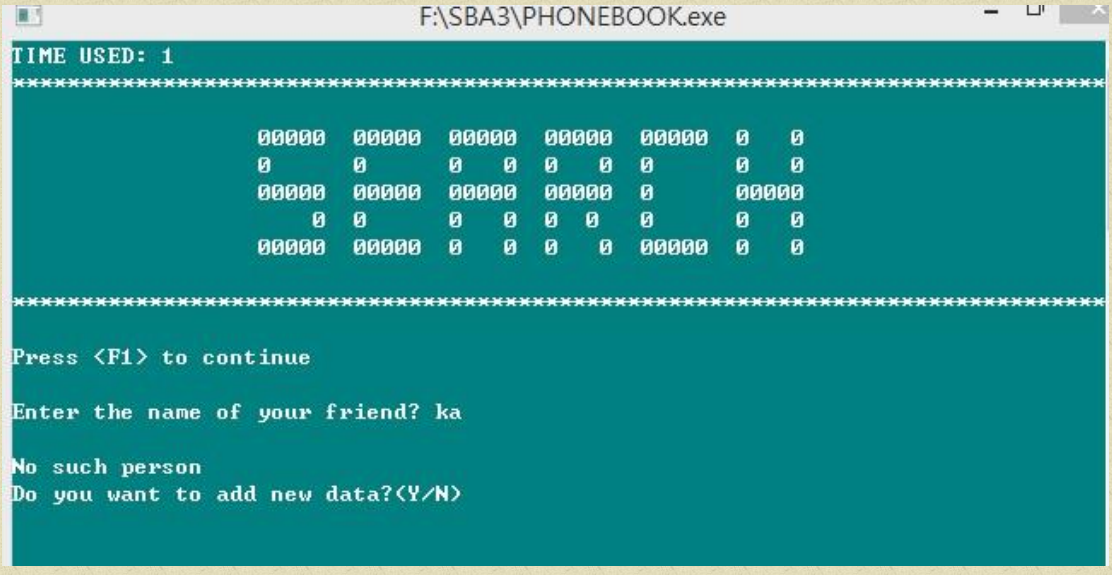

The data of the text file' phone.txt' can successfully read from the program!

Text2 Searching


'Phone.txt'

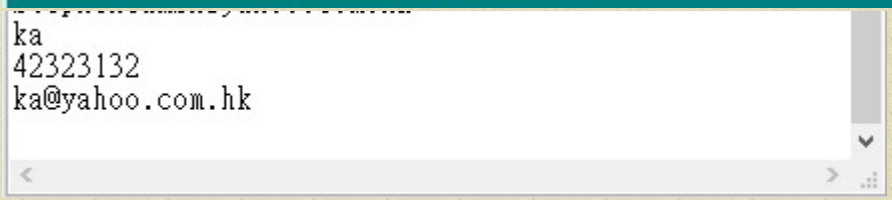



When I enter the name that is appeared in my 'phone.txt', the data can be shown on my program and finish the searching process.



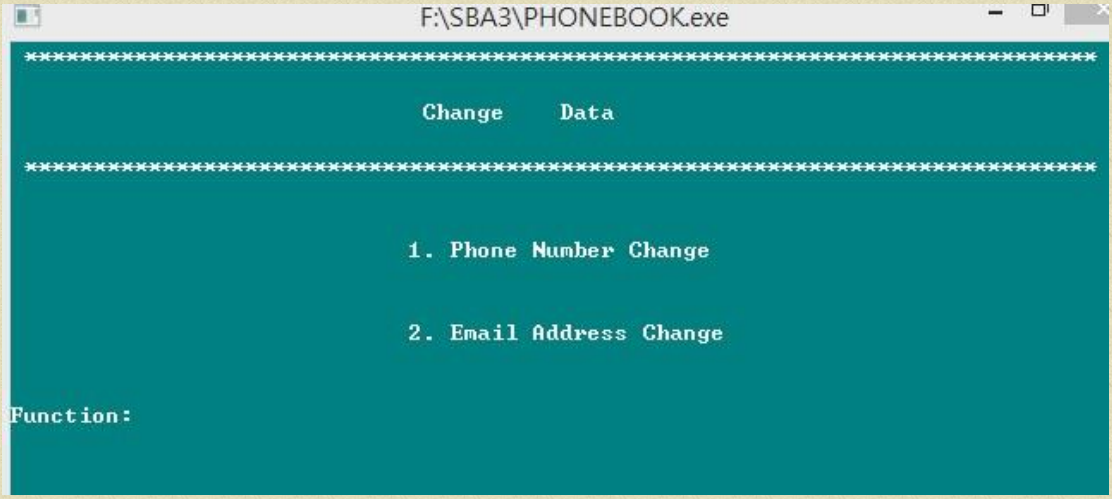

When I input a name that is not appear in my program, the program will return the sentence 'No such person' and ask whether you want to add data or not, if answer is 'Y', the program will move to the add data function.

Test 3 Add data

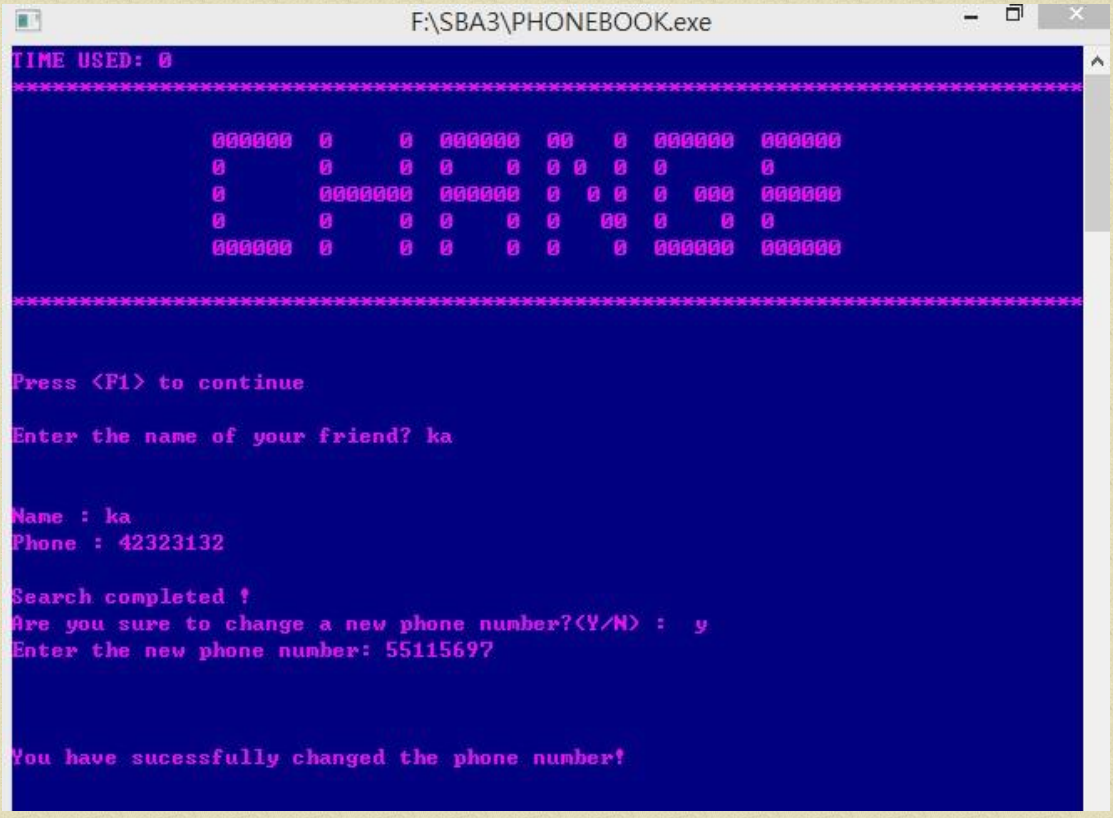

When a new data is added into the phonebook, the new data will write into the 'phone.txt', that mean the data will save in the program. Only the small capital and large capital can be entered and only the phone number with 8-digit phone number can be entered, otherwise the program will return a statement 'invalid data'

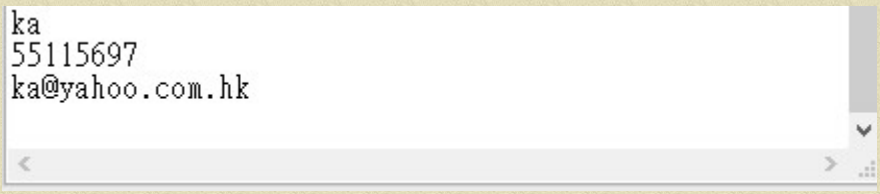
## Test 4 Change data




After the choice no.4 is entered, it will appear a sub-program.

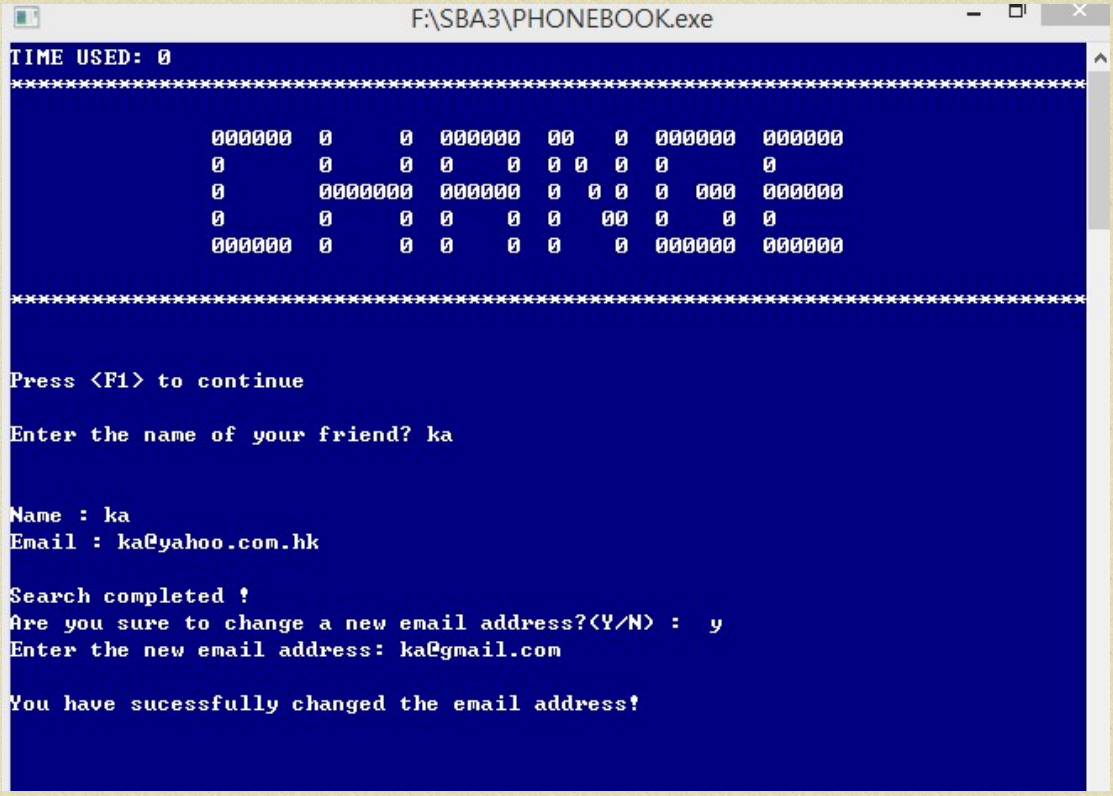
## Change phone number

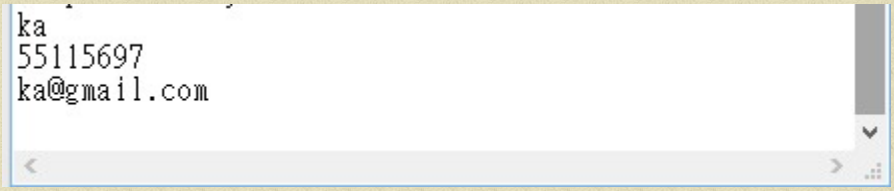





After enter the name of a person, the information of that person can be shown and there will be a question ask whether you want to add a new phone number, if answer is 'Y', you can add a new phone number and the new phone number will then save into the program and the test file.

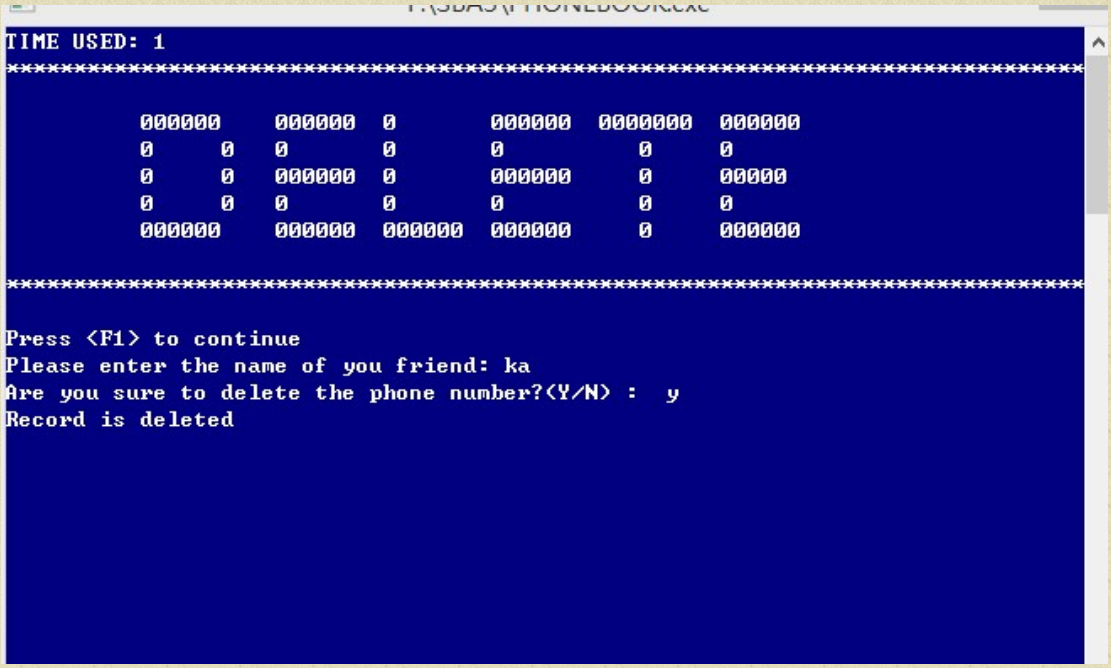
Change e-mail address

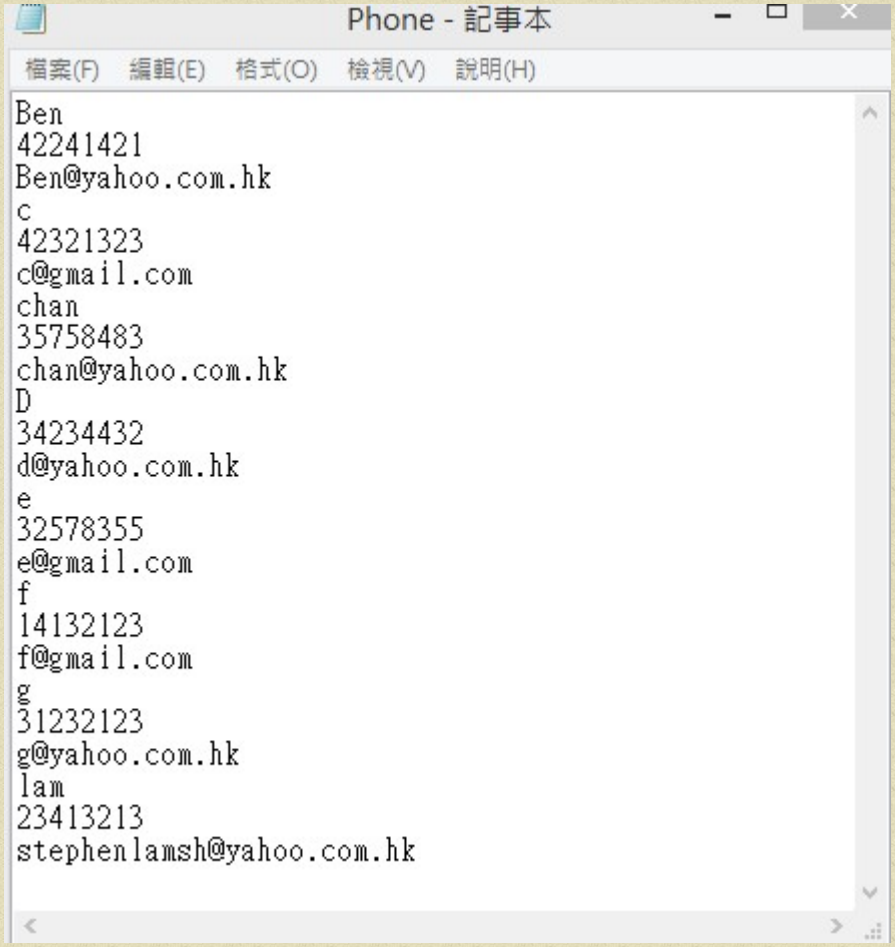



It is as same as the change phone number function.

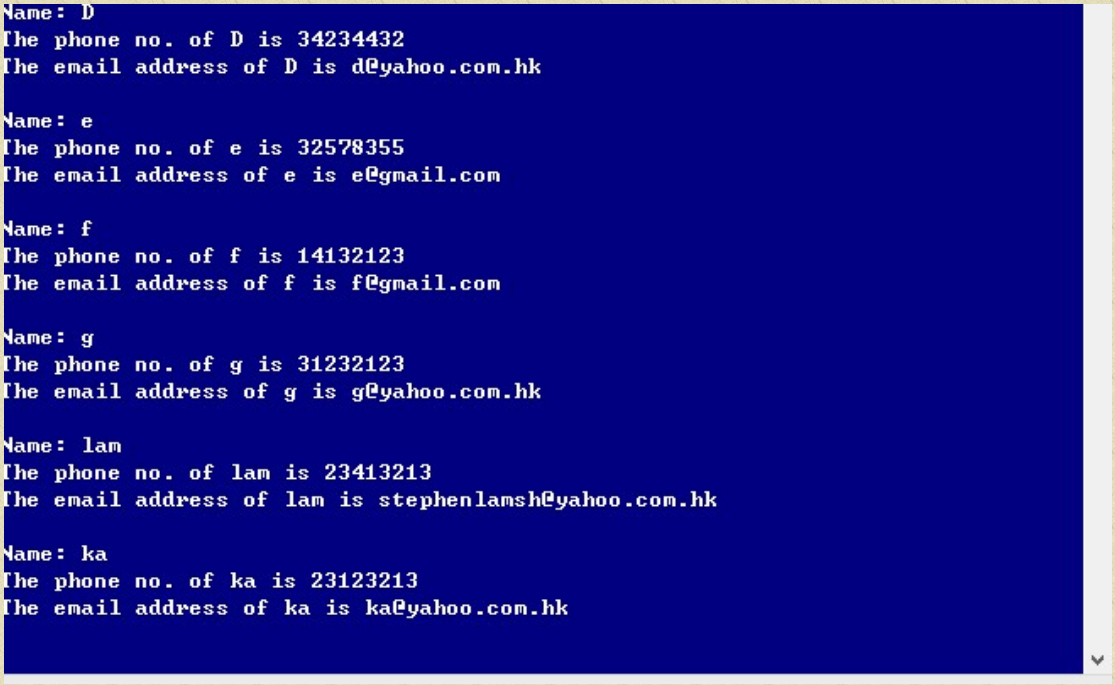
## Test 5 Delete Data

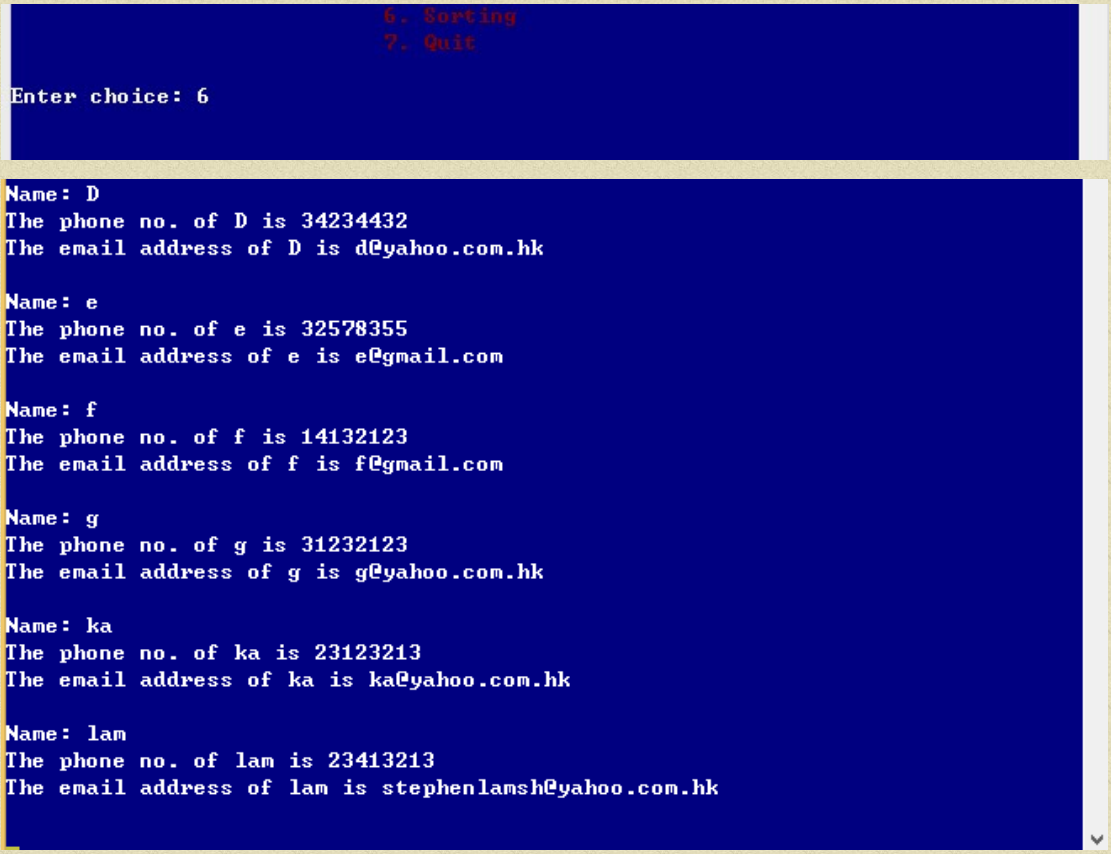





After the name is entered, the data of that person will remove and the data of that person will not appear in the text file.

## Test 6 Sorting






The data can be sorted by the character A-Z after the choice number 6 is entered.

### 3.2 Self evaluation

Actually, I think the program has already performed the minimum requirements of the intended users. The program can satisfy the wants of the users. They can do the function just like the normal phone book directory. Also, I made the phone book become more user-friendly, so it is suitable for the users.

## 4. Conclusion and Discussion

From the implementation process of the program, it shows that the program can run smoothly and it is able to allow the user to search data, change data, add data, delete data and also the sort data. I' ve also made it become more user-friendly, so I think it may be beneficial for the global citizen. However, the interface is a bit bored, so I think that is the only cons

of the whole program.

## 4.1 Reflections

Through doing this SBA, I found that my programming knowledge is enriched since a lot of researches had been done for this program before I started on working it. From the whole process of the program, I have shown positive attitude in learning more about how to improve the function of my programs. I also feel surprised that I have ability to finish the whole program in a given period of time. I feel success after finish this phone book program. I feel happy by writing this program and I feel glad that I can learn more about the programming and the software development during the SBA.

## 4.2 Further Development plans

The following are the plans for further development (If there is any)

*Choose another software which support a better user interface.

*Improve the quality of my program and be more user-friendly

*Add more new functions into my program

*Find out methods to strengthen the functions

# 5. Project Management

## 5.1 Activity Schedule

| Date | Planned schedule | Actual Schedule |
|---|---|---|
| 29/11 | To start my program | Start to write my program |
| 30/11 | Finish writing the procedure readfile and writefile | Finish writing the procedure readfile and writefile |
| 20/12 | Finish all the procedure and start to design my program | Not yet finish to write the whole procedure |
| 30/12 | Testing & Debugging | Finish the whole program and have a testing & debugging |
| 7/1 | Finish the report | Finish the report and the whole SBA. |

## 5.2 Adapting to Changes

At first, I prefer to finish the whole program in one and a half months. However, when I tried to start my program, I found that there are many mistake and bugs in my program. The progress rate starts to slow down since I need to waste times on debugging works, especially for the debugging works on the searching function, so the whole progress is over my expectation. After facing lots of the bugs, I start to understand how to solve the problem and adapt the changes.

Appendix

Acknowledgement

All references

www.hkeaa.edu.hk

http://eclass.cswcss.edu.hk

www.hk.yahoo.com

www.google.com

http://en.wikipedia.org

http://pascal-programming.info/

All source of assistance

Thanks to my dear ICT teachers for assistance – Mr. Chu ☺

Thanks to my cit buddy and best friend- Anson Lo, Bobby Chan and Albert Lai☺

Thanks to my computer☺

Thanks God☺

Source codes

```
program PhoneBook;

Uses Crt;

const    Max = 100;
           datafile = 'Phone.txt';

var    Name : array[1..Max] of string;
         Phone : array[1..Max] of string;
         Email : array[1..Max] of string;
         N, Choice : integer;


procedure ReadFile;
var Infile : text;
       i : integer;
begin
   assign(Infile, datafile);
   reset(Infile);
   i := 0;
   while not eof(Infile) do
      begin
         i := i+1;
         readln(Infile, Name[i]);
         readln(Infile, Phone[i]);
         readln(Infile, Email[i])
      end;
   close(Infile);
   N := i
end;

procedure WriteFile;
var Outfile : text;
       i : integer;
begin
   assign(Outfile, datafile);
```

```
    rewrite(Outfile);
    for i := 1 to N do
        begin
            writeln(Outfile, Name[i]);
            writeln(Outfile, Phone[i]);
            writeln(Outfile, Email[i])
        end;
    close(Outfile)
end;

procedure DisplayData;
var i : integer;
begin
clrscr;
writeln('**********************************************************************
**************');
writeln('                    000000   0       0   000000   0         0   000000   0
0');
writeln('                    0        0       0   0       0   0         0   0       0   0
0');
writeln('                    000000   000000   0       0   0   0   0   000000   0
0');
writeln('                            0   0       0   0       0   0 0 0 0   0       0   0           0');
writeln('                    000000   0       0   000000   00     00   0       0   000000
000000');
writeln('');
writeln('**********************************************************************
**************');
    writeln('List of records: ');
    writeln;
    for i := 1 to N do
        begin
            writeln('Name: ', Name[i]);
            write('The phone no. of ');
            write(Name[i]);
            writeln(' is ', Phone[i]);
            write('The email address of ');
            write(Name[i]);
```

```
            writeln(' is ', Email[i]);
            writeln
        end;
end;

procedure Insertnew;
Label start,finish,timeout;
var NewName,NewEmail,
  NewPhone : string;
    a: integer;
    valid : boolean;
var i : integer;

Begin
      start:
            a:= 0;
            Repeat
            ClrScr;
                  Writeln('TIME USED: ',a);
writeln('*****************************************************************
**************');
writeln('                              00000   00     0   00000   00000   00000   00000');
writeln('                                0      0 0   0   0         0         0     0      0');
writeln('                                0      0   0 0   00000   00000   00000      0');
writeln('                                0      0    00         0   0         0   0       0');
writeln('                              00000   0      0   00000   00000   0     0      0');
writeln('');
writeln('*****************************************************************
**************');

            writeln('Press <F1> to continue');
                        inc(a);
                  Delay(1000);
                     If (a=0) Then Begin
                       GoTo finish;
                     finish:
                     readln;
                     end;
```

```
        Until KeyPressed;

  repeat
  writeln;
      valid := true;
write('Please enter the name of the new record: ');
readln(NewName);
      for i:= 1 to length(Newname) do
      If Not (NewName[i] in ['A'..'Z','a'..'z',' ']) then
          valid := false;
          writeln;
      If Not valid then
      write('Invalid data! ' );
      writeln
      until valid;

  repeat
  writeln;
    valid := true;
  write('Please enter the phone no.: ');
  readln(NewPhone);
      for i := 1 to length(Newphone) do
      If not (Newphone[i] in ['0'..'9',' ']) then
      valid:= false;
      If (length(Newphone) > 8) or (length(Newphone) <8) then
      valid := false;
      writeln;
      If not valid then
      write('Invalid data! ');
      writeln;
      until valid;

write('Please enter the email address: ');
readln(NewEmail);
N := N + 1;
Name[N] := NewName;
Phone[N] := NewPhone;
Email[N] := NewEmail;
```

```
    writefile
end;

procedure Changephone;
Label start,finish,timeout;
var a: integer;
var i,j,k : integer;
var Findname : string;
var Valid, Position : boolean;
var Choice : char;

      Begin
Textbackground(Blue);
clrscr;
Textbackground(Blue);
Textcolor(LightMagenta);
      start:
           a:= 0;
           Repeat
           ClrScr;
                Writeln('TIME USED: ',a);
writeln('**********************************************************************
**************');
writeln('                        000000   0        0   000000   00     0   000000   000000');
writeln('                        0        0        0   0       0  0 0   0   0           0');
writeln('                        0        0000000  000000  0    0 0   0   000   000000');
writeln('                        0        0        0   0       0   0     00   0       0   0');
writeln('                        000000   0        0   0       0   0      0   000000   000000');
writeln('');
writeln('**********************************************************************
**************');
writeln;
           writeln('Press <F1> to continue');
                      inc(a);
                Delay(1000);
                   If (a=0) Then Begin
                     GoTo finish;
                   finish:
```

```
                readln;
                end;
          Until KeyPressed;

begin
  repeat
  writeln;
     valid := true;
  write('Enter the name of your friend? ');
  readln(Findname);
       for i := 1 to length(Findname) do
  if length(Findname) > 25    then
  valid := false;
  if not(Findname[i] in ['A'..'Z','a'..'z',' ']) then
  valid := false;
  writeln;
        if not valid then
           writeln('Invalid data !');
     until valid;
     writeln;

     Position := true ;
      for j := 1 to 30 do
         if (Findname[j] <> Name[i][j] ) and (Findname[j] <> '@' ) then
            Position := false ;

            begin
              for i := 1 to N do
            begin



if upcase(Findname) = upcase(Name[i]) then

  begin

  position := true;
     writeln('Name : ', Name[i]);
```

```
    writeln('Phone : ', Phone[i]);
        writeln;
  write('Search completed !');
  writeln;

 write('Are you sure to change a new phone number?(Y/N) :    ');
 readln(choice);

 if upcase(choice) = 'Y' then
    begin
    repeat
    valid := true;
 write('Enter the new phone number: ');
 readln(Phone[i]);
      for k := 1 to length(Phone[i])do
      If (length(phone[i]) > 8) or (length(phone[i]) < 8) then
            valid := false;

      writeln;
      If not valid then
      write('Invalid data!');
      writeln;
      until valid;
 writeln;
 writeln('You have sucessfully changed the phone number!');
            end;
            end;
   end;
 end;
         If not position then
                    write('No such person');
end;
writefile;
end;


procedure Changeemail;
Label start,finish,timeout;
```

```
var a: integer;
var i,j : integer;
var Findname : string;
var Valid, Position : boolean;
var Choice : char;

Begin
     start:
          a:= 0;
          Repeat
          ClrScr;
               Writeln('TIME USED: ',a);
writeln('**************************************************************
**************');
writeln('                    000000   0        0   000000   00     0   000000   000000');
writeln('                    0        0        0   0       0   0 0   0   0          0');
writeln('                    0        0000000  000000  0   0 0   0   000   000000');
writeln('                    0        0        0   0       0   0     00   0       0   0');
writeln('                    000000   0        0   0       0   0       0   000000   000000');
writeln('');
writeln('**************************************************************
**************');
writeln;
          writeln('Press <F1> to continue');
                    inc(a);
               Delay(1000);
                  If (a=0) Then Begin
                    GoTo finish;
                  finish:
                  readln;
                  end;
          Until KeyPressed;

begin
  repeat
  writeln;
  write('Enter the name of your friend? ');
  readln(Findname);
```

```
        valid := true;
        for i:= 1 to length(Findname) do
        If not (Findname[i] in ['A'..'Z','a'..'z',' ']) then
        valid := false;
   if length(Findname) > 25 then
   valid := false;
   writeln;
         if not valid then
            writeln('Invalid data !');
      until valid;
      writeln;

      Position := true ;
       for j := 1 to 30 do
          if (Findname[j] <> Name[i][j] ) and (Findname[j] <> '@' ) then
             Position := false ;

          begin
           for i := 1 to N do
          begin



if upcase(Findname) = upcase(Name[i]) then

      begin

   position := true;
      writeln('Name : ', Name[i]);
      writeln('Email : ', Email[i]);
         writeln;
    write('Search completed !');
   writeln;
   write('Are you sure to change a new email address?(Y/N) :    ');
   readln(choice);

   if upcase(choice) = 'Y' then
   begin
```

```
  write('Enter the new email address: ');
  readln(Email[i]);
  writeln;
  writeln('You have sucessfully changed the email address!');
        end;
        end;
     end;
  end;
            If not position then
                        write('No such person');
end;
writefile;
end;

procedure changedata;
var choice : integer;
begin
clrscr;
writeln('
**********************************************************************
***** ');
writeln('                                          Change       Data
');
writeln('
**********************************************************************
***** ');
writeln;
writeln('                                          1. Phone Number Change
');
writeln;
writeln('                                          2. Email Address Change
');
writeln;
write('Function: ');
readln(choice);
writeln;
case Choice of
    1:Changephone;
```

```
    2:Changeemail;
    end;
writefile
end;


procedure search;
Label start,finish,timeout;
Var a:Byte;
var i,j : integer;
var Findname : string;
var Valid,Position : boolean;
var choice : char;


Begin
      start:
           a:= 0;
           Repeat
           ClrScr;
                Writeln('TIME USED: ',a);

Writeln('**********************************************************************
**************');
writeln('                          00000   00000   00000   00000   00000   0     0');
writeln('                          0         0         0     0   0     0   0         0     0');
writeln('                          00000   00000   00000   00000   0         00000');
writeln('                                0   0         0     0   0   0     0         0     0');
writeln('                          00000   00000   0     0   0     0   00000   0     0');
writeln('');
writeln('**********************************************************************
**************');

                writeln('Press <F1> to continue');
                  inc(a);
                  Delay(1000);
                     If (i=0) Then Begin
                       GoTo finish;
```

```
                finish:
                readln;
                end;
        Until KeyPressed;

    begin
        repeat
        writeln;
        valid := true;
        write('Enter the name of your friend? ');
        readln(Findname);
        for i := 1 to length(Findname) do

        if Not (Findname[i] in ['A'..'Z','a'..'z',' ']) then
                valid := false
        else
            if length(Findname) > 25 then
                valid := false;

        if not valid then
             writeln('Invalid data !');
        until valid;

        writeln;

    Position := true ;
      for j := 1 to 30 do
        if (Findname[j] <> Name[i][j] ) or (Findname[j] <> '@,#,$,%,^,*,(,)' ) then
            Position := false ;

              begin
                for i := 1 to N do
              begin



if upcase(Findname) = upcase(Name[i]) then
```

```
 begin
     position := true;
     writeln('Name : ', Name[i]);
     writeln('Phone : ', Phone[i]);
     writeln('Email : ', Email[i]);
        writeln;
   write('Search completed !');
        readln;
   write('Do you want to change data?(Y/N) ');
   readln(choice);
   If upcase(choice) = 'Y' then
   ChangeData;
 end;
                end;
                end;
                        If not position then
                      write('No such person');
                      writeln;

       write('Do you want to add new data?(Y/N)');
       readln(choice);
       If upcase(choice) = 'Y' then
       Insertnew;
       end;
end;




procedure DeleteData;
Label start,finish,timeout;
var i, Target : integer;
       FindName : string;
       a: integer;
       choice: char;

Begin
     start:
```

```
        a:= 0;
        Repeat
        ClrScr;
            Writeln('TIME USED: ',a);

writeln('**********************************************************************
**************');
writeln('              000000      000000   0          000000   0000000   000000');
writeln('              0       0   0        0          0              0        0');
writeln('              0       0   000000   0          000000         0        00000');
writeln('              0       0   0        0          0              0        0');
writeln('              000000      000000   000000     000000         0        000000');
writeln('');
writeln('**********************************************************************
**************');
        writeln('Press <F1> to continue');
                    inc(a);
            Delay(1000);
               If (a=0) Then Begin
                 GoTo finish;
               finish:
               readln;
               end;
        Until KeyPressed;

  write('Please enter the name of you friend: ');
  readln(FindName);

  Target := 0;
  for i := 1 to N do
     if upcase(Name[i]) = upcase(FindName)
       then Target := i;
  if Target > 0 then
      begin
      write('Are you sure to delete the phone number?(Y/N) :    ');
readln(choice);
if upcase(choice) = 'Y' then
           begin
```

```
                for i := Target to N-1 do
              begin
                Name[i] := Name[i+1];
                Phone[i] := Phone[i+1];
                Email[i] := Email[i+1]
              end;
       writeln('Record is deleted');
              N := N - 1
              end;
         end;
     writefile;
end;

procedure sorting; {sort for the DisplayData}
var
   k, i : integer;
   temp_name, temp_phone, temp_email : string;
   swap : boolean;
begin
     k := 0;
   repeat
   k := k + 1;
   swap := false;
   for i := 1 to N - k do
       if upcase(name[i]) > upcase(name[i + 1]) then
           begin
              temp_name := name[i];
              name[i] := name[i + 1];
              name[i + 1] := temp_name;
              temp_phone := phone[i];
              phone[i] := phone[i + 1];
              phone[i + 1] := temp_phone;
              temp_email := email[i];
              email[i] := email[i + 1];
              email[i + 1] := temp_email;
              swap := true
              end;
              writefile;
```

```
    until (k = N - 1) or not swap;
end;



begin {main program}

Textbackground(lightCyan);

    Repeat
        ReadFile;
        Clrscr;



Textcolor(LightMagenta);
writeln('                    ************************************************
');
writeln('                    * **   *   ** ***** *       ***** ***** ** **   ***** *
');
writeln('                    *   ** * **    **    *       *       *    * * * *    **    *
');
writeln('                    *    ** **    ***** ***** ***** ***** *    *   ***** *
');
writeln('                    ************************************************
');
Textcolor(Lightgreen);
writeln('');
write(' Here is a program for you to search for the phone no. of the people you want to find.
');
        writeln('');
        writeln('');
Textcolor(Yellow);

writeln('*********************************************************************
**************');
Textcolor(White);
        writeln('                                                    Menu');
```

```
        writeln('');
        Textcolor(Yellow);

writeln('**********************************************************************
**************');
Textcolor(Red);
        writeln('');
        writeln('                                        1. Show all the data');
        writeln('                                        2. Search');
        writeln('                                        3. Add Data');
        writeln('                                        4. Change Data');
        writeln('                                        5. Delete Data');
        writeln('                                        6. Sorting');
        writeln('                                        7. Quit');
        writeln('');
Textcolor(White);
        write('Enter choice: ');
        readln(Choice);
        writeln;

        case Choice of
            1 : DisplayData;
            2 : Search;
            3 : InsertNew;
            4 : ChangeData;
            5 : DeleteData;
            6 : sorting
        end;

        readln
    until Choice = 7;
    clrscr;
    writeln;
    writeln(' 00000 00000 00000 0000      0000    0      0 00000      0' );
    writeln(' 0          0      0 0      0 0      0    0      0    0 0    0             0' );
    writeln(' 0 000 0      0 0      0 0        0 0000        0      00000      0' );
    writeln(' 0      0 0      0 0      0 0      0    0      0      0      0              ' );
    writeln(' 00000 00000 00000 0000      0000        0      00000      0' );
```

```
    readln;
    WriteFile
end.
```

# END

Thank you for paying attention to my report! :)