Cheung Sha Wan Catholic Secondary School

Information & Communication Technology

School Based Assessment

Phone Book System

Content

1. Objective and Analysis

1.1 The Background and the Situation

Nowadays, the smart phone has become one of the most important things towards people. Phone Call has been replaced by the online communication software. Many people think that the phone call function is not necessary to them. It is because people can chat online and texting. So, the phone call has been ignored by the public. Although the social networking site has launched the mobile apps to the public, the online phone call still need the phone numbers and other necessary data. As the phone numbers are mostly stored in the SIM card inside mobile phone, people may find difficult to maintain the data. As a result, I think that using the program to search and modify the phone book system can be more efficient. It is because using the computer can be collated the phone numbers into an more details categories.

1.2 The Sub Problems

In this program, there are some sub-problems inside the program. For example, determining the searching algorithm and the sorting algorithms of the phone book. So, I decided to put this at the two major problems of this program. It is because the searching algorithm will determine the whole program processing. As the main purpose of a phone book system is to search the particular phone within a huge set of data. Therefore, I considered the searching algorithm is the main body of the whole system. As a result, I think that the searching algorithms will affect the whole program operation to become efficient or not. In the perspective of the sorting, this is regarded as the element which will affect the user-friendly of the whole system. So, I have to make the decision of choosing the suitable field to become the sorting target. Due to having the right field of sorting group, the phone book system will become more user-friendly and being general accepted.

1.3 Target Users & User Requirements

The target user of this phone user system is the computer user. It is because the phone book system is only support the computer. The phone book system is decided for the user with basic computer skill. It is because the phone book system supports the computer only. So, the user needs to have some basic computer knowledge when they are using it. Moreover, the user may require with some Basic English reading skill. It is because the statements inside the program are all written in English. So, the user needs to equip with Basic English reading skill and understand some Basic English vocab. So, I think that this will be the user requirements.

1.4 Considerations

In this program, there are some considerations inside the phone book system. As the algorithms of the system need to make some decision making, it will become

one of the considerations. Also, the user-interface may be another consideration. It is because the user-interface and the wording of the program may be important to the user-friendliness of the program. If the expansion of the program and the design of the program are not in well structure, the usability of the program will be affected. Therefore, these will be the most important consideration of the whole program.

1.5 Software Analysis

In this program, I have used the Pascal software to write this phone book system. It is because the Pascal is a simple program language. Also, the Pascal is similar with the English writing. So, using the Pascal to write can be more easily to complete the program. Also, the Pascal contain a striated program structure. So, it is more suitable for the program writing beginner to develop the program system.

In the Pascal software, I choose the dev-Pascal to write the program coding. It is because the dev-Pascal is free of charge and the user-interface of the dev-Pascal is user friend. So, I decide to use the dev-Pascal to do the project.

2. Design and Implementation

2.1 The User interface

When I was writing the program, I tend to use the Command Line Interface (CLI) instead of the Graphical User Interface (GUI). It is because the command line interface is more suitable for the experience users. It is because the user has to remember the commands. So, the CLI is more suitable for the experiment users and the users with some ICT knowledge. Furthermore, using the CLI can help to save the resources because the photos are not needed for the CLI.

2.2 Modularization

In this program, it has divide into four sub-programs. There are edit, create, delete and sorting. In this program, the sub-programs are operating under the searching which is the main program of this phone book system. It is because the edit and the delete function can be call after the searching has been done. So, the searching function acts as a important character in this program. As a result, the searching procedure will be reused in this program for several times.

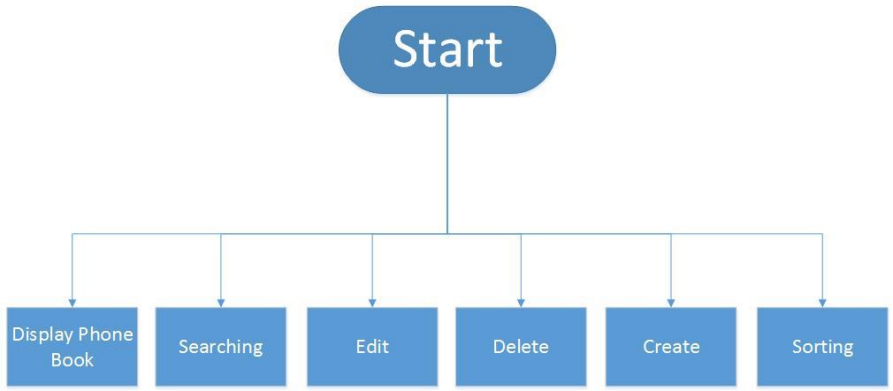


Fig.1 Main Menu

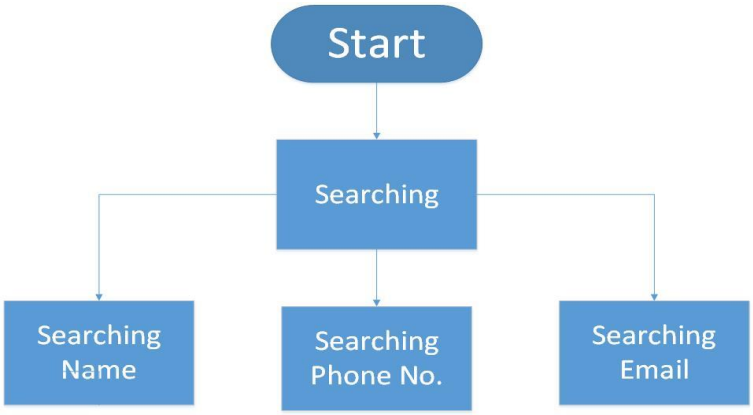


Fig.2 Searching Menu

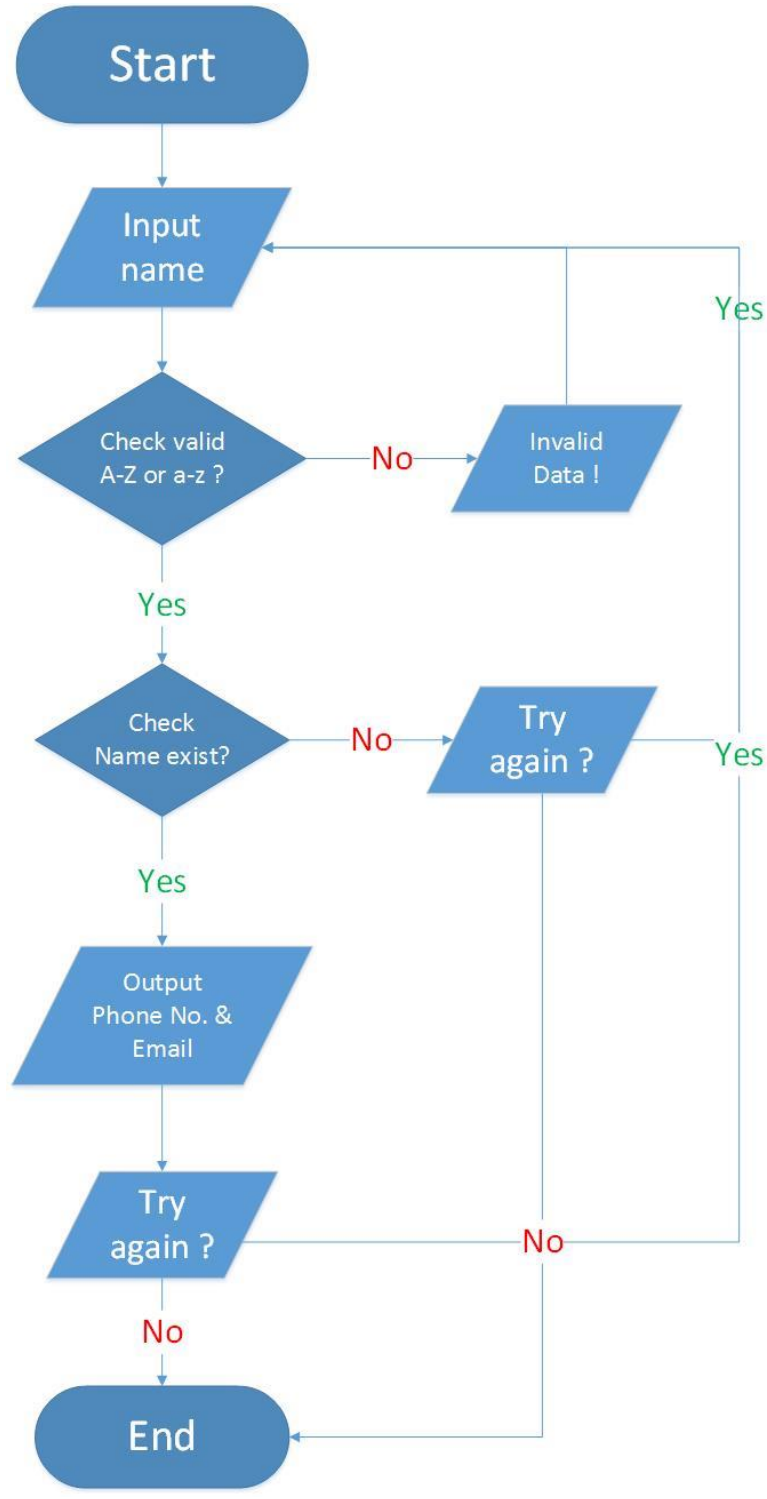


Fig.3 Searching Name

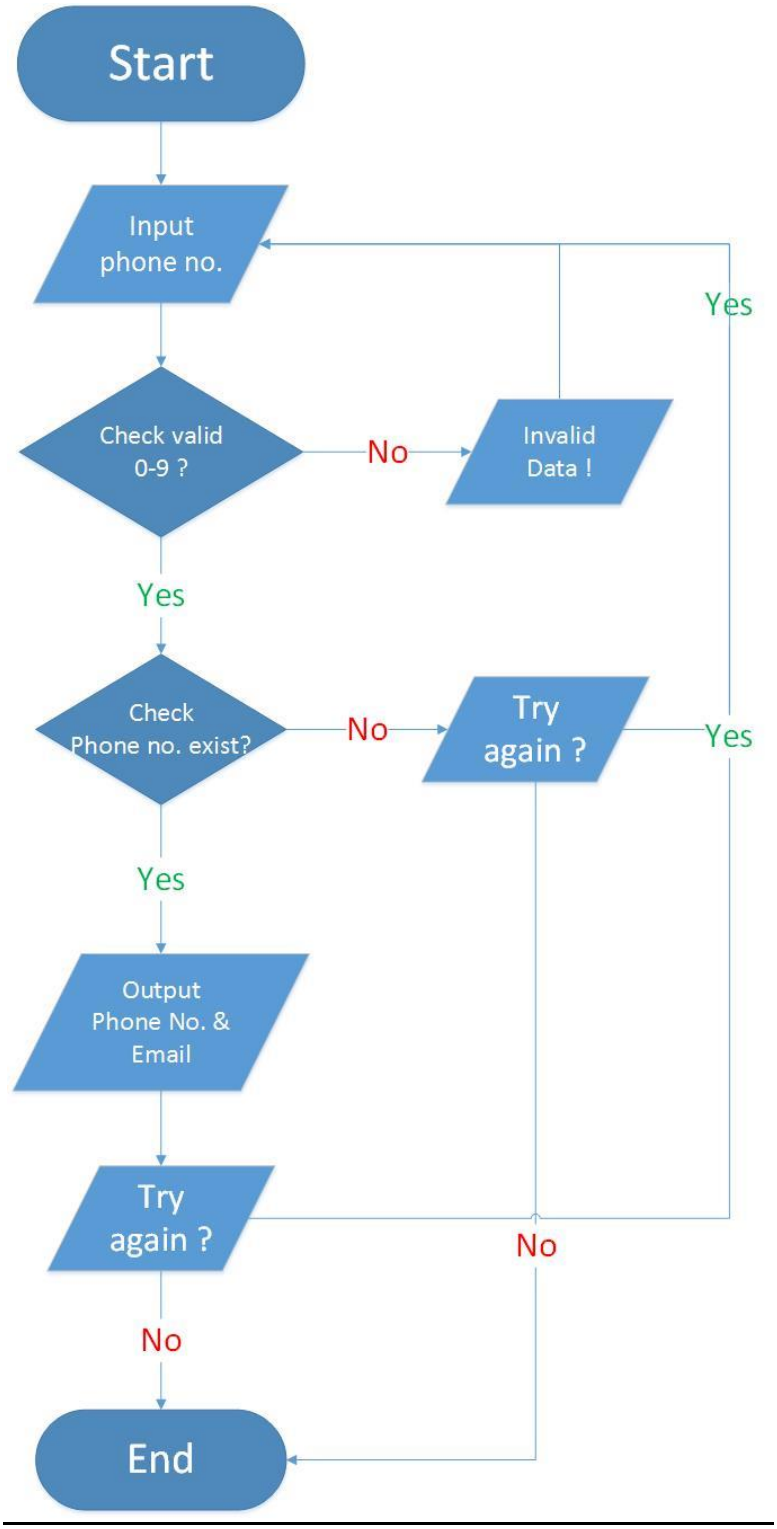


Fig.4 Searching Phone No.

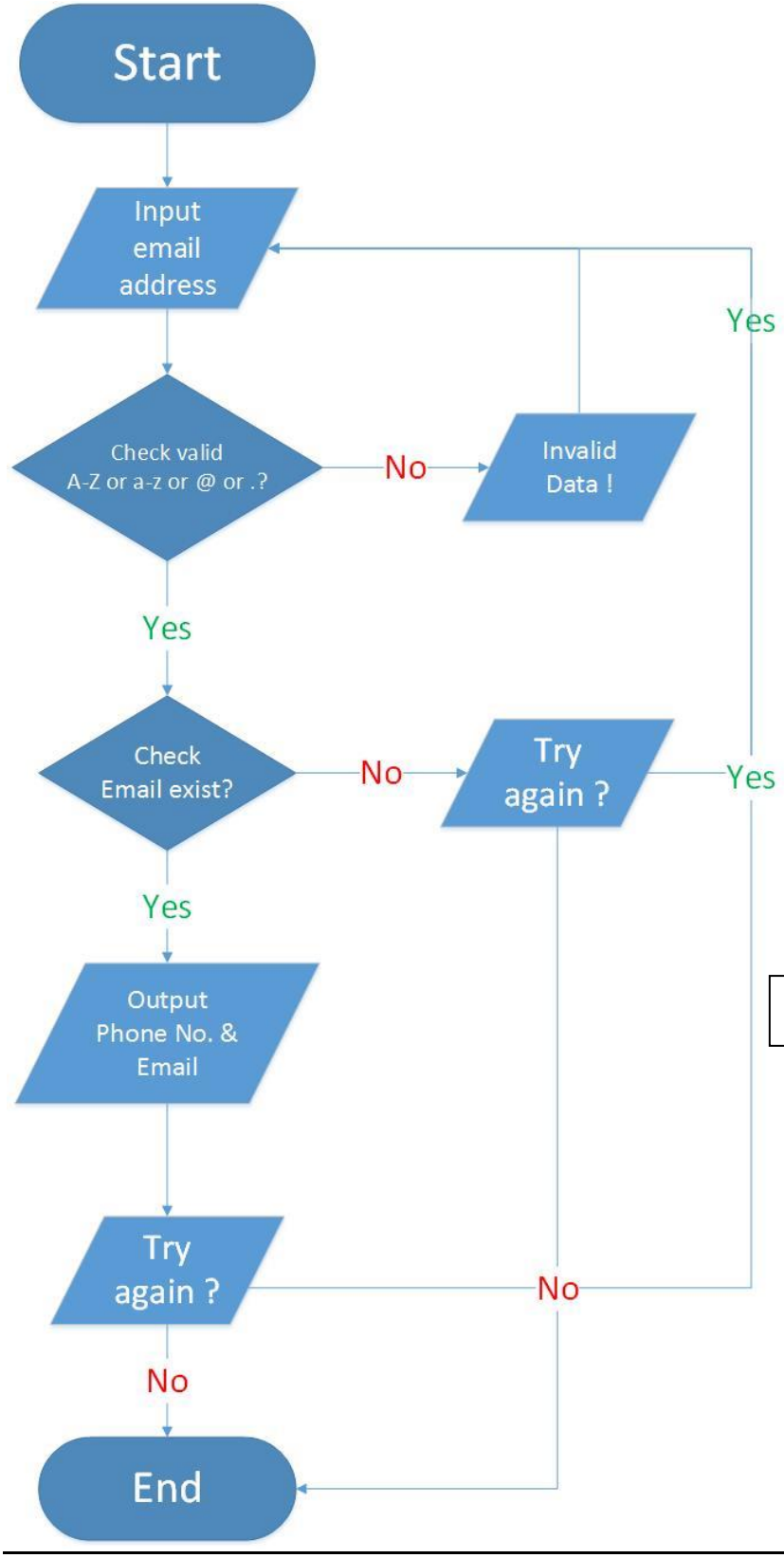


Fig.5 Searching Email Address

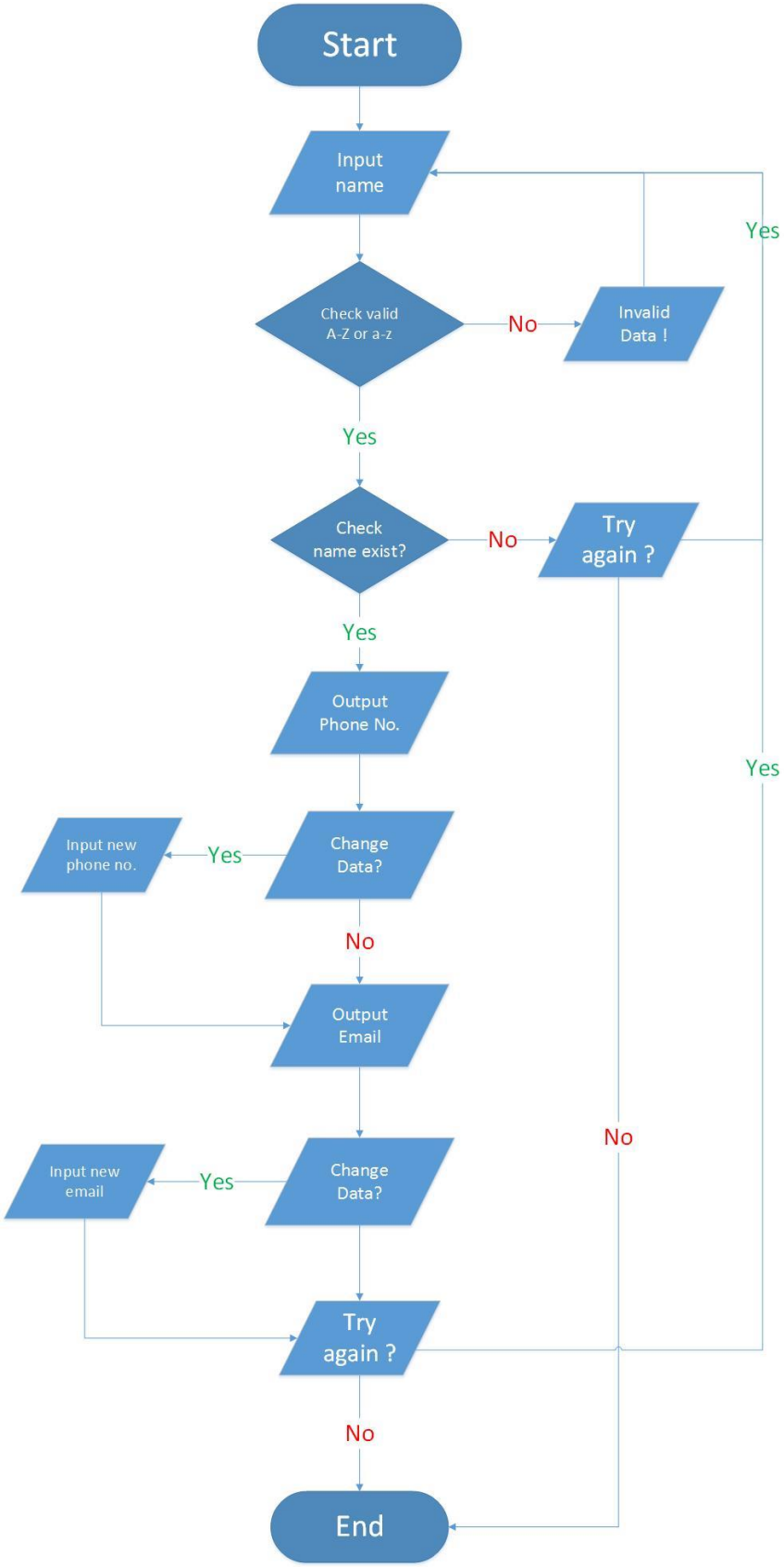


Fig.6 Edition of Phone Book

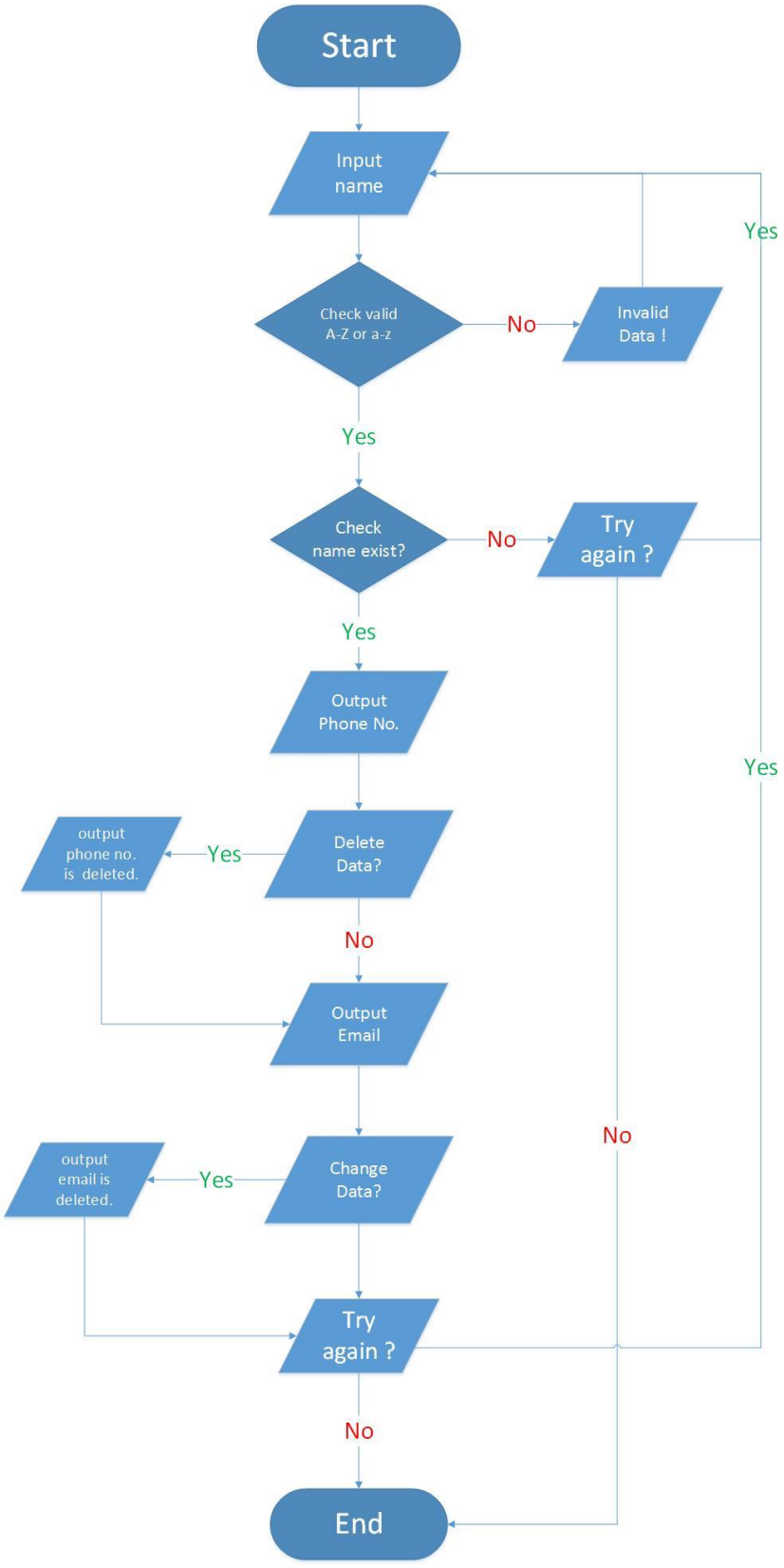


Fig.7 Delete the record of Phone Book

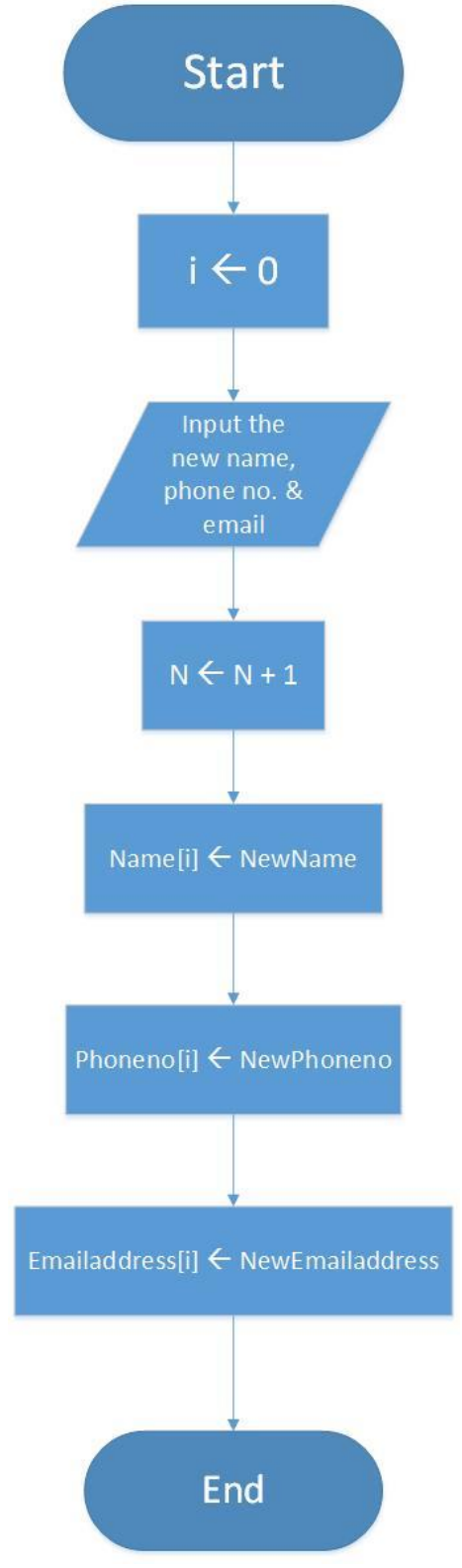


Fig.8 Create the new record of Phone Book

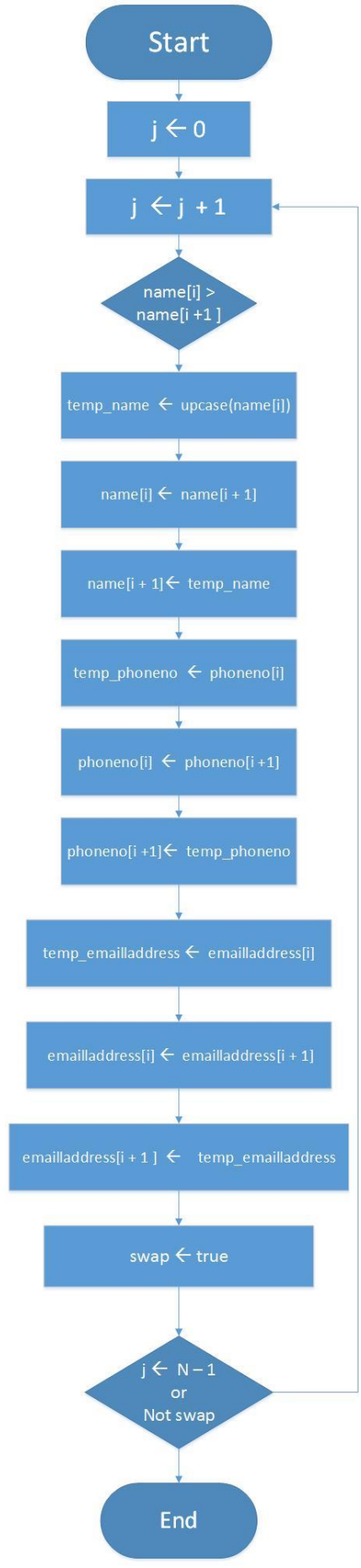


Fig.9 Sorting

2.3 The Data Structure

In this program, I use the file to store the data of the phone numbers, names and email address. Also, in the program, I used three array to store the above three data. It is because I think that using the file to store the data can be easily to maintain. As the data in the file can be edit and delete easily. So, I tend to use the file to store the data instead of the record. It is using the file can let the program become simple and more easily to debugging and data validation. So, the using the file format to store the data can be easily to control by the program beginner. So, I prefer using the file to the record.

Besides that, using the array to store the data is more efficient. It is because there are three main data information. As to reduce the redundant of listing the data record, using the array can prevent this problem. It is because when the array is used, the target data can be found by changing the indicator inside the specific array. Therefore, the program can access the data faster and directly. The searching procedure can be operated more efficient due to the redundant program coding is not exist. So, I tend to use the array to store the data inside the program.

2.4 Constants and Limitations

In this program the maximum size of the data record is 100. It is because the data record of the phone book may not be too large. So, the scale of the array is set at the amount of 100. When consider to some of the users work in the business community, the maximum of the data size can be increased by changing the maximum number. It is because the array is set from the beginning data to the maximum size of data. So, the scale of the record size and the capacity of the array can be determined by the career of the user easily. Also, the maximum number can be changed according to the demand of the customer and the program user. Therefore, I think that it is reasonable and flexible to set the maximum record size at 100. Moreover, I think that it is also easily to change and reset the data scale of the phone book and the array. As I have mentioned before, the end of the array is set as the maximum size of the phone book record.

2.5 Data Control

In order to reduce the input error, there is a validation check after the input of the data in each sub-program. It is because the typing error cannot prevent when the user are inputting a large variety of data at the same time. So, using the validation check can help to minimize the input error.

In the validation check, there are some requirements for checking the input data. First, in the searching name part, if the input doesn't satisfy with the requirement of the either small capital or the large capitals, the program will only output the command of the invalid data. Also, the spacebar is accepted in the

checking. When the input passes the above checking, the program will start the searching function to find whether the input name exists or not. So, I think that using the above requirements and criteria to check the input data can prevent the user input some incorrect data format. When the validation checking is used in the program, the program will be more efficient to deterrent the incorrect data input.

Besides having the validation checking in the input name, the input phone number also need to pass the validation check. It is because the user may not only input the numbers in this column and they may enter so irrelevant integer. So, using the validation check can help to prevent this problem. So, in this column, only the input within 0 to 9 can pass the checking. After the checking function is pass, the searching will be called. If the input data is invalid, then the program will only output the statement of invalid data. Therefore, it can help to reduce the time of searching the incorrect format data. It is because the validation checking help to filter the incorrect and the unsuitable format of data.

Apart from these, the email also using the same method of the validation checking to check for the input data. In this checking, the inputs which satisfy with some requirements can be passed. The requirements are the small capital or the large capital , the symbol of '@' and the digit from 0 to 9. So, if the input data doesn't satisfied with the above requirements, the program will only output the statement of invalid data. As the user id of the email address can be created by the users individually. Thus, it is difficult to set some strict checking requirements. So, I think that it will be reasonable to set these basic requirements for the validation check.

For the above three validation checks will be used in the searching procedure. Only the validation checking in name will be reused in other procedure like the edit, delete function. So, these validation checking will take the responsibility of checking the input data.

2.6 The Operations of Each Module

In this program, there are some main modules in this program. They are the searching module, the edit module, delete module, sorting module and the create module.

First, in the searching module, it consist three searching methods in this module. The user can determine to use which methods in their searching. As the Data control section has mentioned that the searching method are involve an validation check to check for the input data. In the searching name module, the input data which violate the requirements, like contain some symbols and numbers, the program will output the statement of invalid data and ask the user to input the data again. If the input data passes the validation check, the searching algorithms will operate and search for the phone record. When the data pass the above two checking, the phone record,

like the phone number and the email address. After all the process, the question try again will be asked. The whole module will be ended when the input is not satisfied with Y or y .

In the searching phone and searching email modules apply the same searching and the validation checking. But, only the validation checking criteria is not the same with the searching name one. In the phone number searching , only the nine digits which is 0-9 are accepted. And in the email address searching, the criteria of the validation checking is all the alphabet in both capital and small letter. Also, the symbol of the full-stop and @ will be accepted. It is because this are the base requirements and the most common cases and formats of the email address and the phone number.

Besides from the searching modules, the edit module will be another important modules in this program. It is because the phone book user always need to modify the record of the phone book. So, in this module, the validation check used in the name searching module will be reused in this module. After the validation checking, the program will find out the existence of the input data. When the data is found, then the program ask for updating the phone book record. Only when the user input the choice of y or Y , the program will provide the changing space to the user. When the user finished editing the phone record, the question ''try again ''will be asked.

Apart from these the delete function will be perform similar with the edit function. Only the statement output will be changed. As the data record is deleted, the file and the program will only displace some spaces and the blank column to represent the empty data.

On the other hand, the creation of the data record is different with the others. It is because after the data is input, only the name of the phone book record will change into the capital letters and the others will remain unchanged. So, the data will add to the record file and the total number of the phone record will be increased.

Last but not least , the sorting algorithm in the program helps to sort the phone book record. In this sorting, the sorting field is using the name field. It is because the user may always use the name to search for the phone record. So, the whole record file will be sorted by using the name field. According the ASCII code , it is printed from the capital letter A to Z and then the small capital letter a to z. So, after using the up case function, the name will write in the capital letter and then the sorting will base on the ASCII Code to do it.

2.7 The Algorithms applied

In this program, the sorting algorithms are used. In this phone book, I tend to use the bubble sort. As the phone book is always list according to the capital. So, using the

bubble sort can help to group the name record into the correct order. In this sorting, the target group is the names of the friends and others. So, the whole sorting will be carried out under the name. It is because people always use the name as the main searching category. So, I think sorting name in the correct order will be convenient for the general user. It is because the user doesn't need to adapt the unfamiliar searching interface when they are using this phone book system. Therefore, I believe that using the name as the sorting target is user-friendly and efficient for the user to use.

Besides that, in this program, I also used the liner searching to search for the phone book data. It is because using the Liner Searching will be faster than the Binary Searching. As the common phone book record is in a small scale, the liner searching is affordable in this kind of situation. Compare with the liner searching and binary searching, the speed of searching the data within the small scale of data is nearly the same. So, there is no an absolute advantage of using the binary searching in this program. Therefore, I think that using the liner searching in this phone book is reasonable and feasible.

2.8 User guide of the Program

In this program, there are some variable in this program to facilitate the operation of the program. In this program, the array of name[i], phoneno[i] and emailaddress[i] are used for storing the data of the phone record. Then, the variable [i] is used as the counter in these array.

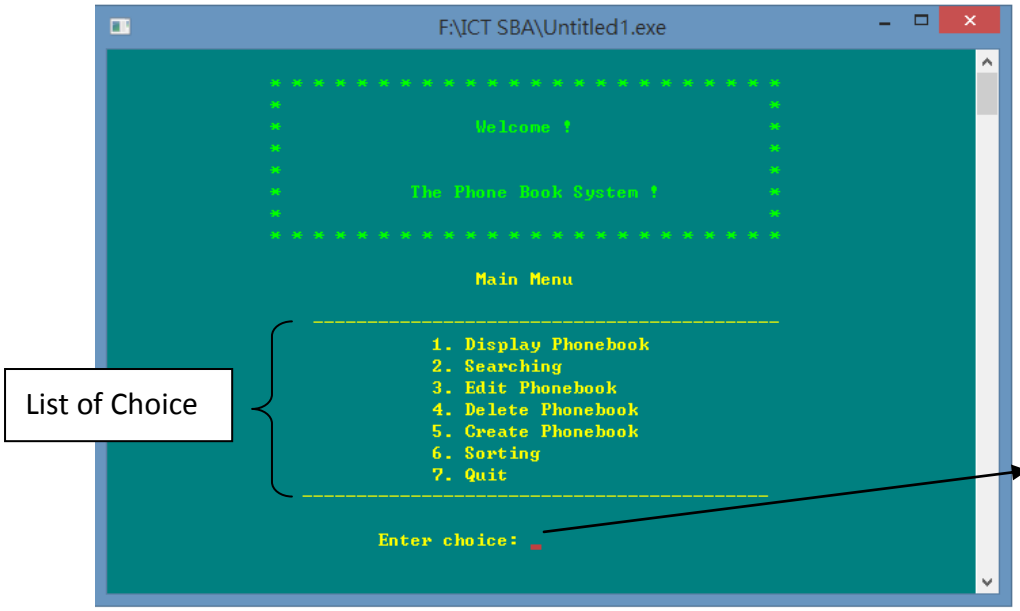


Fig.10 Main Menu

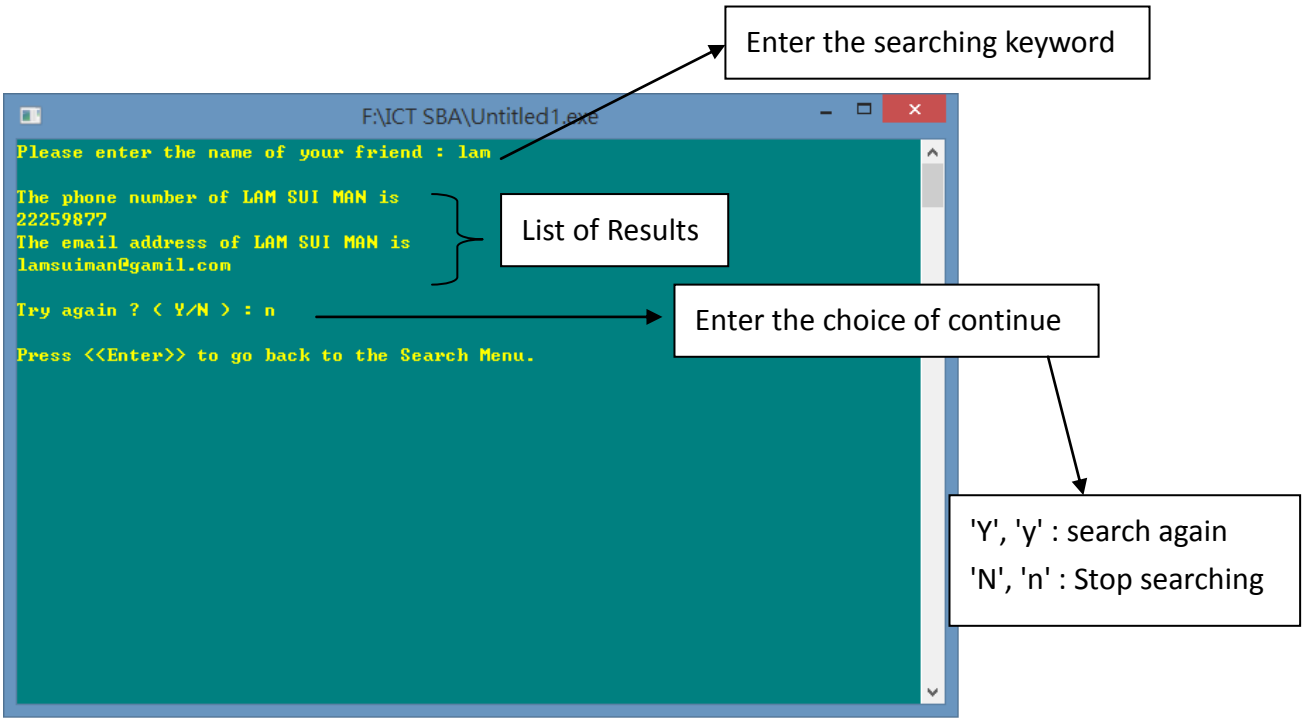


Fig.11 Searching Name

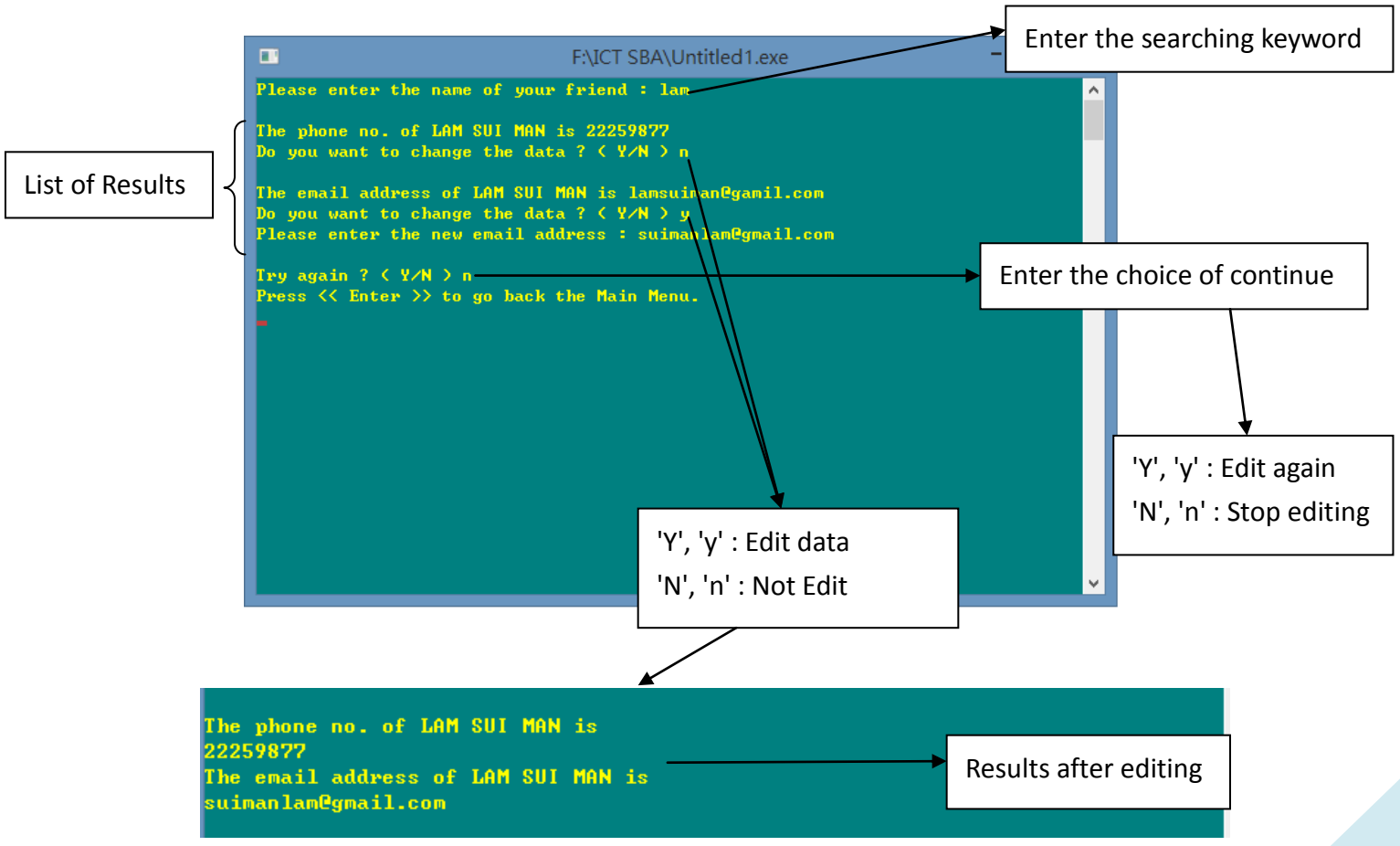


Fig.12 Edition & Result

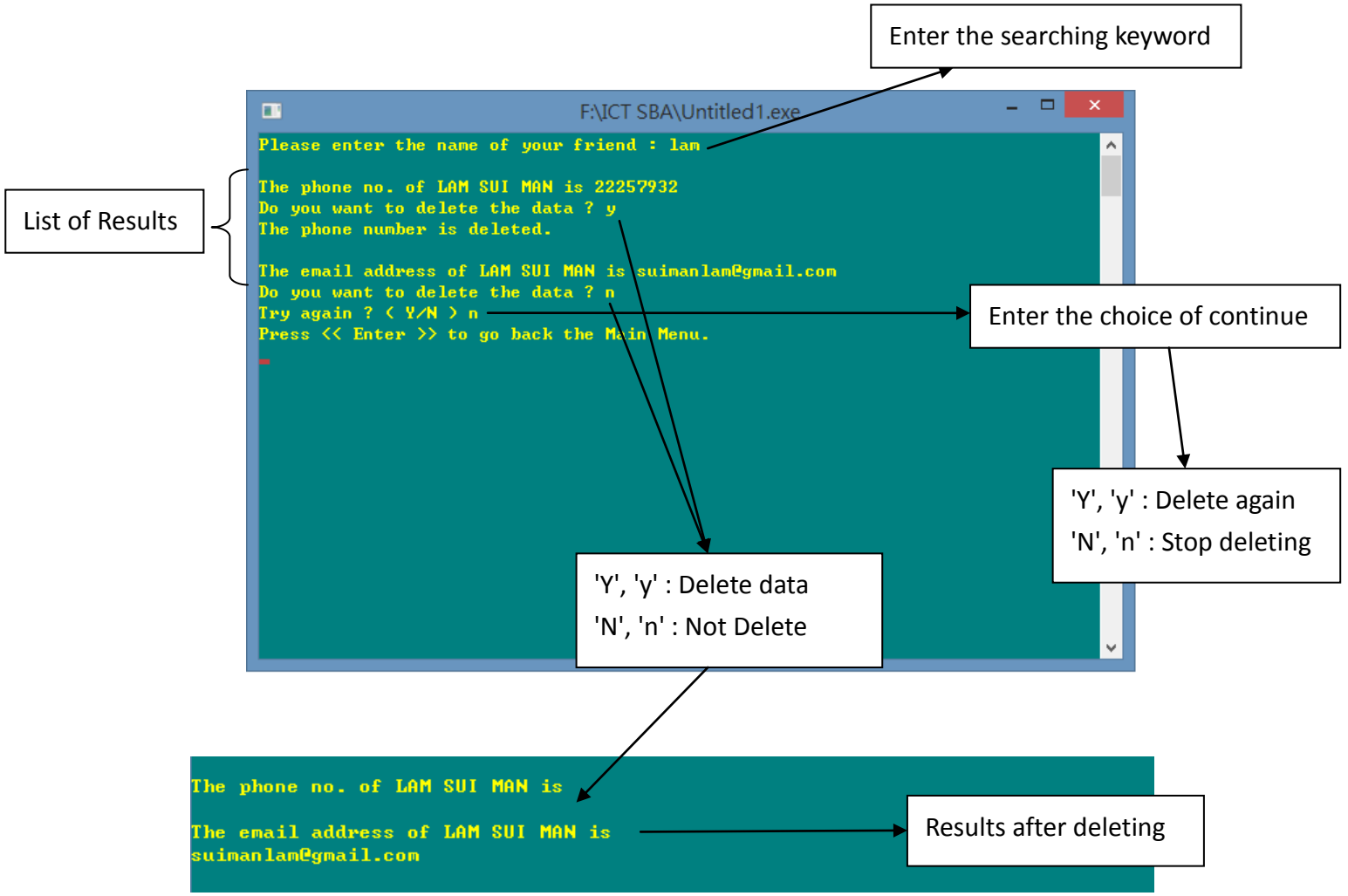
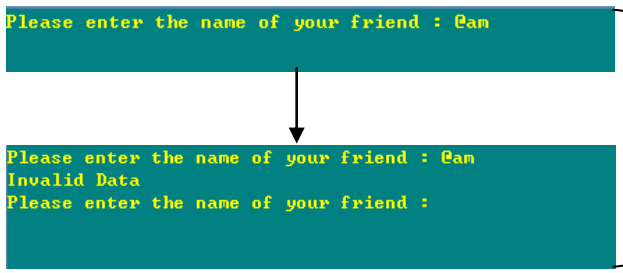


Fig.13 Delete & Result

With reference to the above screen shooting, the basic operation of the whole program is introduced. As there are some pointers and some basic description about some important column, the user can know more about how the screen is operated and the meaning of the sentences.

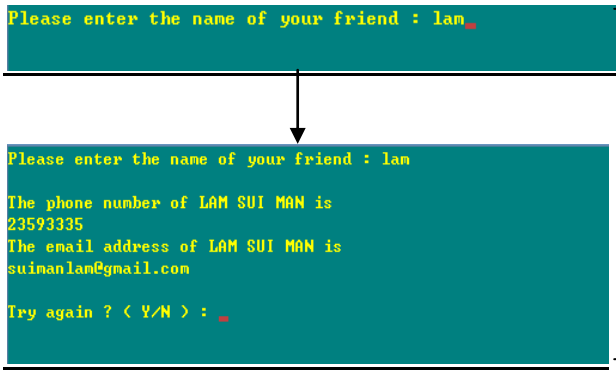
3. Testing and Evaluation

3.1 Test the Program



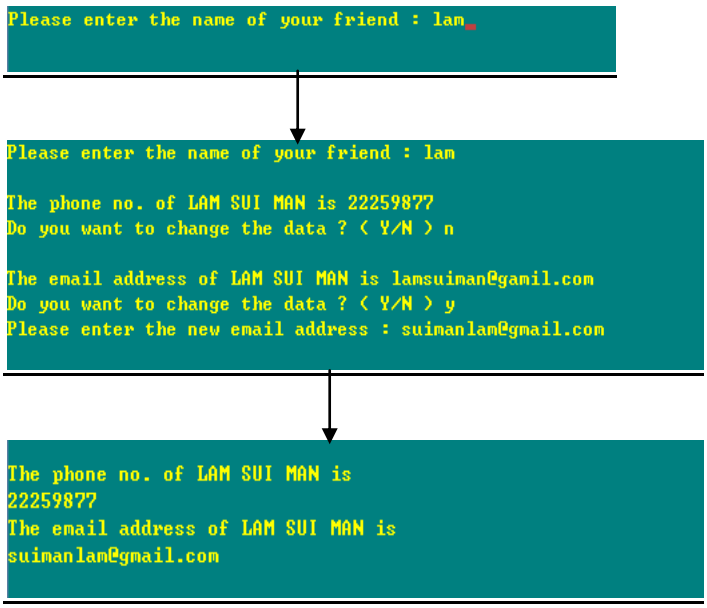
Test 1 : Searching Name
(Invalid Case)

Fig.14 Test No. 1



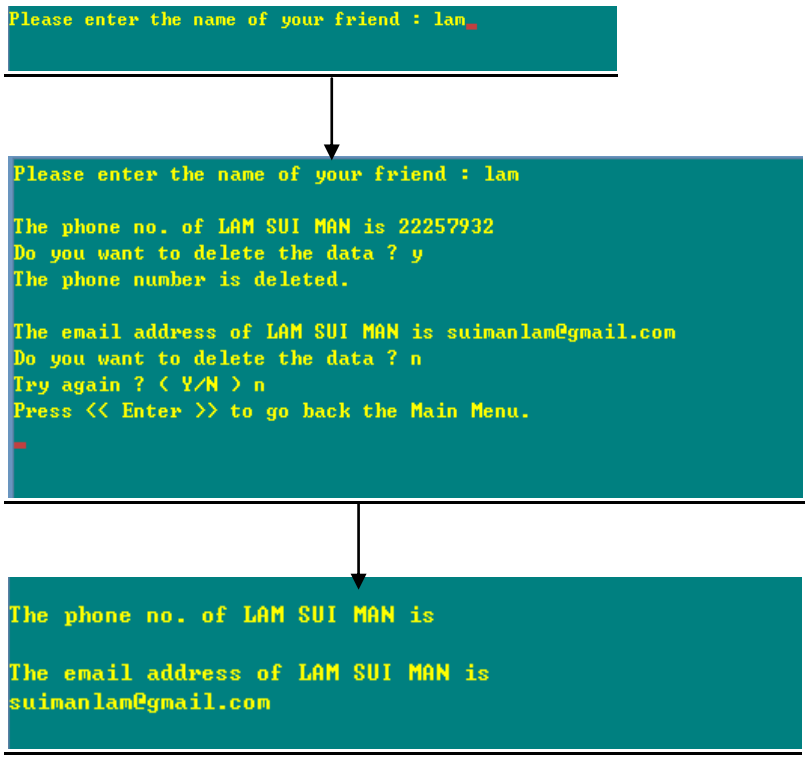
Test 2 : Searching Name
(Valid Case)

Fig.15 Test No. 2



Test 3 : Editing Email
(Positive Result)

Fig.16 Test No. 3



Test 4 : Delete Phone Number
(Positive Result)

Fig.17 Test No. 4

3.2 Evaluate the Program

I think that the program is seems to be user-friendly. Most of the basic functions inside the program are similar with the real phone book system used in the mobile device. But, the display of the program is not well designed. So, the display of the program is the flaw of the program. Therefore, the program cannot be identify as the most user-friendly and most functional phone book system

4. Conclusion and Discussion

4.1 Conclusion

After finishing the whole project, it seems that the program match with majority of the aims and objectives of the project. In this program, it has provide an user-friendly interface. Although the interface seems to be friendly and readable, the font size of the wordings inside the program cannot increased. For me, I think it is an technological problems and may be out of the syllabus. So, it becomes one of the problems and obstacles in the program coding.

Besides that, the presentation of the program is not well enough as I had drafted before. When I was designing the program, I think that it can list out all the results first. Then, state the modify statement at the bottom. But, in don't come true. It is because there are consist some technology problem and some structure design. So, the interface layout did not apply in this program.

4.2 Skills Obtained from the Project

After doing this project, I have known more about the program design and the program writing. It is because when I has to create a new program from the beginning. As there are not reference for this project. So, I need to find out some more reference from the Internet to deal with the program design. After doing this program, I have developed the design skills of the program. It is because when I was writing the program, I have to consider the user interface and the other program printing.

4.3 Difficulties Encountered during the Implementation

When I was doing this project, the most difficult parts in printing the iteration of the program. It is because looping of the program will be affected the workability of the whole program. So, while deciding the loop body and the choose of looping, I have to compared with several kinds of iteration methods. Finally, I tend to use the repeat loop and the for loop to be main iteration of the phone book system. As the repeat-until loop don't contain some restrict and the condition contain more freedom. So, using the repeat-until loop as the main loop in the procedures. As a result, after finishing the project, I have more deeply understanding in the iteration of the program. So, I can determine the use of iteration in the program.

4.4 Future Development

In the future, I want to improve my program in terms of font size. It is because the font size is not much user friendly at this moment. It is because the eyesight of each user may be different and the user may contain some eye illness. So, in the future, I want to develop the program in the way of the font size as to take care of the users. So, the font size in the program will be planned to improve in the future.

Moreover, the user-interface of the program will be planned to be changed. It is because the program cannot list out all the results of searching in the particular of procedure, like edit and the delete. So, I am going to redevelop the user-interface of the program of these two procedures. As these are the important elements of the phone book system. As a result, I am going to improve these fields of the program.

5. Project Management

5.1 Activity Schedule

Date	Task to be done
May - 2015	Choice of Topic
July - 2015	Background research, Define the objectives, Propose Functions
August - 2015	Design of Solution
October - 2015	Implementation
November - 2015	Testing & Evaluation
December - 2015	Conclusion & Discussion , Final Report

Appendix

1. Eclass F.5 Pascal Ex.18 & 20
2. Wikipedia Dev-Pascal
   https://en.wikipedia.org/wiki/Dev-Pascal
3. Wikipedia Pascal
   https://en.wikipedia.org/wiki/Pascal_%28programming_language%29
4. Wikipedia Telephone Directory
   https://en.wikipedia.org/wiki/Telephone_directory
5. ICT Textbooks Program Development (D1, D2)
6. ICT Textbooks Compulsory 3

Source Code

```
program phonebook;
Uses Crt;

const Max = 100;

var name : array[1..max] of string;
       phoneno : array[1..max] of string;
       emailaddress : array[1..max] of string;
       N, Choice : integer;

Procedure readfile;
var f : text;
       i : integer;
begin
   assign(f , 'phonebook.txt');
   reset(f);
   i := 0;
   while not eof(f) do
      begin
         i := i + 1;
         readln(f, name[i]);
         readln(f, phoneno[i]);
         readln(f, emailaddress[i]);
     end;
       N := i;
       close(f);
end;

Procedure WriteFile;
var Outfile : text;
               i    : integer;
begin
     assign( Outfile, 'phonebook.txt');
     rewrite(outfile);
     for i := 1 to N do
          begin
```

```
            writeln(Outfile, Name[i]);
            writeln(outfile, Phoneno[i]);
            writeln(Outfile, emailaddress[i]);
        end;
   close(outfile)
end;

Procedure display_phonebook;
var i : integer;
begin
    Clrscr;
    writeln('          ------------------------------------------------      ');
    writeln;
    writeln('                          My Phone Book                        ');
    writeln;
    writeln('          ------------------------------------------------      ');
    for i := 1 to N do
      begin
        writeln;
        writeln('The phone no. of ', name[i], ' is ' );
        writeln(phoneno[i]);
        writeln('The email address of ',name[i] , ' is ' );
        writeln(emailaddress[i]);
        writeln;
      end;
writeln('Press << Enter >> to go back the Main Menu. ');
end;

procedure searching_name;
var i : integer;
      inname : string;
      found , valid : Boolean;
      choice : char;
begin
repeat
Clrscr;
      repeat
          valid := true;
```

```
            write('Please enter the name of your friend : ');
            readln(inname);
               for i := 1 to length(inname) do
                  if Not (inname[i] in ['A'..'Z','a'..'z',' ']) then
                     valid := false;
                  if Not valid then
                     writeln ('Invalid Data');
       until valid ;
writeln;
found := false;
     begin
          for i    := 1 to N do
            if copy(upcase(name[i]),1, length(inname)) = upcase(inname) then
               begin
                  found := true;
                  writeln('The phone number of ', name[i] , ' is ');
                  writeln(phoneno[i]);
                  writeln('The email address of ' , name[i] , ' is ');
                  writeln(emailaddress[i]);
                  writeln;
               end;
     end;
     if Not found then
           writeln('The name is not exist ! ');

write('Try again ? ( Y/N ) : ');
readln(choice);
until Not (choice in ['Y' , 'y' ]);
writeln;
writeln ('Press <<Enter>> to go back to the Search Menu. ');
end;

procedure searching_phone;
var i : integer;
       inphone : string;
       found , valid : Boolean;
       choice : char;
begin
```

```
repeat
Clrscr;
        repeat
           valid := true;
           write('Please enter the phone number of your friend : ');
           readln(inphone);
              for i := 1 to length(inphone) do
                 if Not (inphone[i] in ['0'..'9']) then
                      valid := false;
                 if Not valid then
                      writeln ('Invalid Data');
      until valid ;
writeln;
found := false;
        begin
             for i := 1 to N do
                 if copy((phoneno[i]),1,length(inphone)) = (inphone) then
                      begin
                           found := true;
                           writeln('The phone number of ', name[i] , ' is ');
                           writeln(phoneno[i]);
                           writeln('The email address of ' , name[i] , ' is ');
                           writeln(emailaddress[i]);
                      end;
        end;
        if Not found then
             writeln('The phone no. is not exist ! ');

write('Try again ? ( Y/N ) : ');
readln(choice);
until Not (choice in ['Y' , 'y' ]);
writeln;
writeln ('Press <<Enter>> to go back to the Search Menu. ');
end;

procedure searching_emailaddress;
var i : integer;
      inemail : string;
```

```
        found , valid : Boolean;
        choice : char;
begin
repeat
Clrscr;
        repeat
            valid := true;
            write('Please enter the email address of your friend : ');
            readln(inemail);
               for i := 1 to length(inemail) do
                    if Not (inemail[i] in ['A'..'Z','a'..'z','@','.']) then
                        valid := false;
                    if Not valid then
                        writeln ('Invalid Data');
        until valid ;
writeln;
found := false;
            for i    := 1 to N do
               if copy(upcase(emailaddress[i]),1, length(inemail)) = upcase(inemail) then
                  begin
                      found := true;
                      writeln('The phone number of ', name[i] , ' is ');
                      writeln(phoneno[i]);
                      writeln('The email address of ' , name[i] , ' is ');
                      writeln(emailaddress[i]);
                  end;
        if Not found then
            writeln('The email is not exist ! ');

write('Try again ? ( Y/N ) : ');
readln(choice);
writeln;
until Not (choice in ['Y' , 'y' ]);
writeln;
writeln ('Press <<Enter>> to go back to the Search Menu. ');
end;
```

```
procedure searching;
var
    choice : integer ;

begin
repeat
Clrscr;
    writeln;
    writeln('                                Search Menu                          ');
    writeln;
    writeln('                          ----------------------------------------');
    writeln;
    writeln('                             1.   Searching Name                   ');
    writeln('                             2.   Searching Phone No.              ');
    writeln('                             3.   Searching Email Address          ');
    writeln('                             4.   Quit                             ');
    writeln;
    writeln('                          ----------------------------------------');
    write('                        Enter choice: ');
    readln(Choice);
    writeln;
    case Choice of
          1 : searching_name;
          2 : searching_phone;
          3 : searching_emailaddress;
    end;
readln;
until choice =    4;
writeln('Press << Enter >> to go back the Main Menu. ');
end;

Procedure edit_phonebook;
var i : integer;
      inname : string;
      found ,valid : Boolean;
      ans: char;
begin
repeat
```

```
Clrscr;
    repeat
        valid := true;
        write('Please enter the name of your friend : ');
        readln(inname);
            for i := 1 to length(inname) do
                if Not (inname[i] in ['A'..'Z','a'..'z',' ']) then
                    valid := false;
                if Not valid then
                    writeln ('Invalid Data');
    until valid ;
writeln;
found := false;
      begin
          for i := 1 to N do
           begin
                if copy(upcase(name[i]),1, length(inname)) = upcase(inname) then
                    begin
                        found := true;
                        write('The phone no. of ', name[i] , ' is ' );
                        writeln(phoneno[i]);
                        write('Do you want to change the data ? ( Y/N ) ');
                        readln(ans);
                        If ans in ['Y', 'y'] then
                            begin
                                write('Please enter the new phone no. : ');
                                readln(phoneno[i]);
                            end;
                        writeln;
                        write('The email address of ', name[i] , ' is ' );
                        writeln(emailaddress[i]);
                        write('Do you want to change the data ? ( Y/N ) ');
                        readln(ans);
                        If ans in ['Y', 'y'] then
                            begin
                                write('Please enter the new email address : ');
                                readln(emailaddress[i]);
                                writeln;
```

```
                                end;
                        end;
                end;
        end;
if Not Found    then
     writeln('The name is not exist ! ');

write ('Try again ? ( Y/N ) ');
readln(ans);
until Not (ans in ['Y' , 'y' ]);
writeln('Press << Enter >> to go back the Main Menu. ');
WriteFile;
end;

Procedure delete_phonebook;
var i : integer;
       Found, valid : Boolean;
       inname : string;
       ans : char;
begin
repeat
Clrscr;
   repeat
     valid := true;
       write('Please enter the name of your friend : ');
       readln(inname);
       for i := 1 to length(inname) do
         if Not (inname[i] in ['A'..'Z','a'..'z',''']) then
            valid := false;
       if Not valid then
          writeln ('Invalid Data');
       until valid ;
writeln;
found := false ;
     begin
          for i := 1 to N do
             begin
                if copy(upcase(name[i]),1,length(inname)) = upcase(inname) then
```

```
                begin
                    found := true;
                    write('The phone no. of ', name[i] , ' is ' );
                    writeln(phoneno[i]);
                    write('Do you want to delete the data ? ');
                    readln(ans);
                     If ans in ['Y', 'y'] then
                        begin
                           phoneno[i] := '';
                           writeln('The phone number is deleted. ');
                        end;
                    writeln;
                    write('The email address of ', name[i] , ' is ' );
                    writeln(emailaddress[i]);
                    write('Do you want to delete the data ? ');
                    readln(ans);
                       If ans in ['Y', 'y'] then
                        begin
                           emailaddress[i] := '';
                           writeln('The email address is deleted. ');
                           writeln;
                        end;
                end;
            end;
    end;
    if Not Found    then
        writeln('The name is not exist ! ');

write ('Try again ? ( Y/N ) ');
readln(ans);
until Not (ans in ['Y' , 'y' ]);
writeln('Press << Enter >> to go back the Main Menu. ');
WriteFile;
end;

procedure create_phonebook;
var NewName, NewPhoneno, NewEmailaddress : string;
      i : integer;
```

```
begin
Clrscr;
   i := 0;
   write('Please enter the name of the new record: ');
   readln(NewName);
   write('Please enter the phone no. of the new record: ');
   readln(NewPhoneNo);
   write('Please enter the email address of the new record : ');
   readln(NewEmailaddress);
   N := N + 1;
   Name[i] := upcase(NewName);
   Phoneno[i] := NewPhoneno;
   Emailaddress[i] := NewEmailaddress;
WriteFile;
end;

procedure sorting;
var
     i , j : integer;
     temp_name, temp_phoneno, temp_emailladdress : string;
     swap : boolean;
begin
     j := 0;
repeat
     j := j + 1;
     swap := false;
      for i := 1 to N - j do
          if name[i] > name[i+1] then
              begin
                   temp_name := upcase(name[i]);
                   name[i] := name[i + 1];
                   name[i + 1] := temp_name;
                   temp_phoneno := phoneno[i];
                   phoneno[i] := phoneno[i + 1];
                   phoneno[i + 1] := temp_phoneno;
                   temp_emailladdress := emailaddress[i];
                   emailaddress[i] := emailaddress[i + 1];
                   emailaddress[i + 1] := temp_emailladdress;
```

```
                swap := true;
            end;
until (j = N - 1) or not swap;
WriteFile;
end;

Procedure welcome;
begin
  clrscr;
  TextColor(lightgreen);
  writeln;
  writeln('                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * ');
  writeln('                    *                                                     * ');
  writeln('                    *          Welcome !                                  * ');
  writeln('                    *                                                     * ');
  writeln('                    *                                                     * ');
  writeln('                    *    The Phone Book System !                           * ');
  writeln('                    *                                                      * ');
  writeln('                    * * * * * * * * * * * * * * * * * * * * * * * * * * * * ');
end;

Procedure bye;
begin
  clrscr;
  TextColor(white);
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln;
  writeln('                 ____________________________________________________ ');
  writeln('                 |* * * * * * * * * * * * * * * * * * * * * * * * * *|');
  writeln('                 |*                                                   *|');
  writeln('                 |*          Thank You                                *|');
  writeln('                 |*                                                   *|');
  writeln('                 |*      Hope to see you                              *|');
  writeln('                 |*                                                   *|');
```

```
  writeln('                    |*                  again soon!              *|');
  writeln('                    |*                                           *|');
  writeln('                    |* * * * * * * * * * * * * * * * * * * * * * *|');
  writeln('       |_____________________________________________|');
  readln;
end;

begin
Readfile ;
Textbackground(lightcyan);
repeat
welcome;
writeln;
TextColor(yellow);
writeln('                              Main Menu                          '    );
writeln;
writeln('                          ----------------------------------------');
writeln('                                1. Display Phonebook                  ');
writeln('                                2. Searching ');
writeln('                                3. Edit Phonebook');
writeln('                                4. Delete Phonebook');
writeln('                                5. Create Phonebook');
writeln('                                6. Sorting');
writeln('                                7. Quit');
writeln('                         ----------------------------------------');
writeln;
write('                                Enter choice: ');
readln(Choice);
writeln;
case Choice of
        1 : display_phonebook;
        2 : searching;
        3 : edit_phonebook;
        4 : delete_phonebook;
        5 : create_phonebook;
        6 : sorting;
end;
readln;
```

```
until choice =    7;
clrscr;
bye;
end.
```