Content

Chapter 1 Introduction

1.1 Background

As I am the i.t student in my school, I am asked to design a phonebook program to store

the contact information of our schoolmates. With those stored contact information, school can

inform the students even they are graduated about the school events like BBQ.

It can also used for graduated student to find their friends which are lose track of.

Chapter 2 Design

2.1 brief description

In the past, phonebook just contain the names and phone number

But nowadays technology has becoming more and more advanced, people can contact each other via different other ways like email.

So, I am going to set up a phonebook program with different function. It means that in the program, not only are phone number and names, email and months of birth can also be added.

2.2 General Function

There are one main program which contains 4 functions .Users can choose each function they want to use at beginning page

Display Data Function

The display data function can let users to check the data that they saved before and see whether it need to be changed or not .They can use it as a reference to contact people.

Change Data Function

The change data function is for user to change the old or invalid data after checking the data by using display data function .

Delete Data Function

The delete data function is used to delete data that they do not need to use or wrong data after checking it by display data function. it can help to release the storage space.

Add Data Function

The add data function is used to add the data that user want to store. Users can store contact name, phone number, email and month of birth by using this function .

2.3 Design of User Interface

- **Main menu Screen**

```
**************************************************************************
  *                  Menu                                                  *
  *                                                                        *
  *              1. Display Data                                           *
  *                2. Change Data                                          *
  *                3. Delete Data                                          *
  *                  4. Add Data                                           *
  *                   5. Quit                                              *
**************************************************************************
  Enter choice:
```

- **Display Screen**

```
**********************************************************************
*                          Menu                                      *
*                                                                    *
*                    1. Display Data                                 *
*                     2. Change Data                                 *
*                     3. Delete Data                                 *
*                     4. Add Data                                    *
*                     5. Quit                                        *
**********************************************************************
Enter choice: 1
```

Name	Telephone	Email	MOB
Lam Ching ching	54039599	handsome@gmail.com	03
Chan hoi Fu	67373998	abcde@hmail.com	03
Hwang yu shan	53997612	on8plus1@ymail.com	09

- **Change Data Screen**

```
**************************************************************************
*                    Menu                                                *
*                                                                        *
*                1. Display Data                                         *
*                  2. Change Data                                        *
*                  3. Delete Data                                        *
*                  4. Add Data                                           *
*                  5. Quit                                               *
**************************************************************************
   Enter choice: 2
Please enter the ContactName of your friend:
```

- **Delete Data Screen**

```
**********************************************************************
*                    Menu                                            *
*                                                                    *
*              1. Display Data                                       *
*                2. Change Data                                      *
*                3. Delete Data                                      *
*                  4. Add Data                                       *
*                    5. Quit                                         *
**********************************************************************
   Enter choice: 3
Please enter the ContactName of your friend:
```

- **ADD Data Screen**

```
**********************************************************************
*                    Menu                                            *
*                                                                    *
*              1. Display Data                                       *
*                2. Change Data                                      *
*                  3. Delete Data                                    *
*                    4. Add Data                                     *
*                      5. Quit                                       *
**********************************************************************
   Enter choice: 4
Please enter the ContactName of the new record:
```

Chapter 3 implementation

3.1 Brief description

The Dev-pascal is our choice to implement the NSS student-subject allocation system which I mentioned above. We are going to make a source program and compile it to an executable program.

To begin with, I will design the simply structure of the program by considering the procedure in the program. Besides the procedure, the sample output from above should be well-considered too as I want to make the appearance of the program better.

After that, I will start on producing the program by using the Dev-pascal with different program codes, procedures, functions, etc.

Last but not least, I will have an explanation on the execution of the program so that I can see clearly about the program flow.

3.2 Program coding

In the term of program code, I will introduce some of those and all of the program code will show in my complete program in the Appendix Section, so you may refer to the Appendix Section 1 for the reference.

1) Array

The following parallel array will be used to store the ContactName, TelePhone , Email and month of birth respectively in the text file of Phone.txt.

```
ContactName : array[1..Max] of string;
    TelePhone : array[1..Max] of string;
    Email : array[1..Max] of string;
    MOB : array[1..Max] of integer;
```

2) Text file

Example:

```
assign(Infile, datafile);
 reset(Infile);
```

A text file can help to store data from the program so that more data can be stored. Besides, the program can also read and get data from the text file.

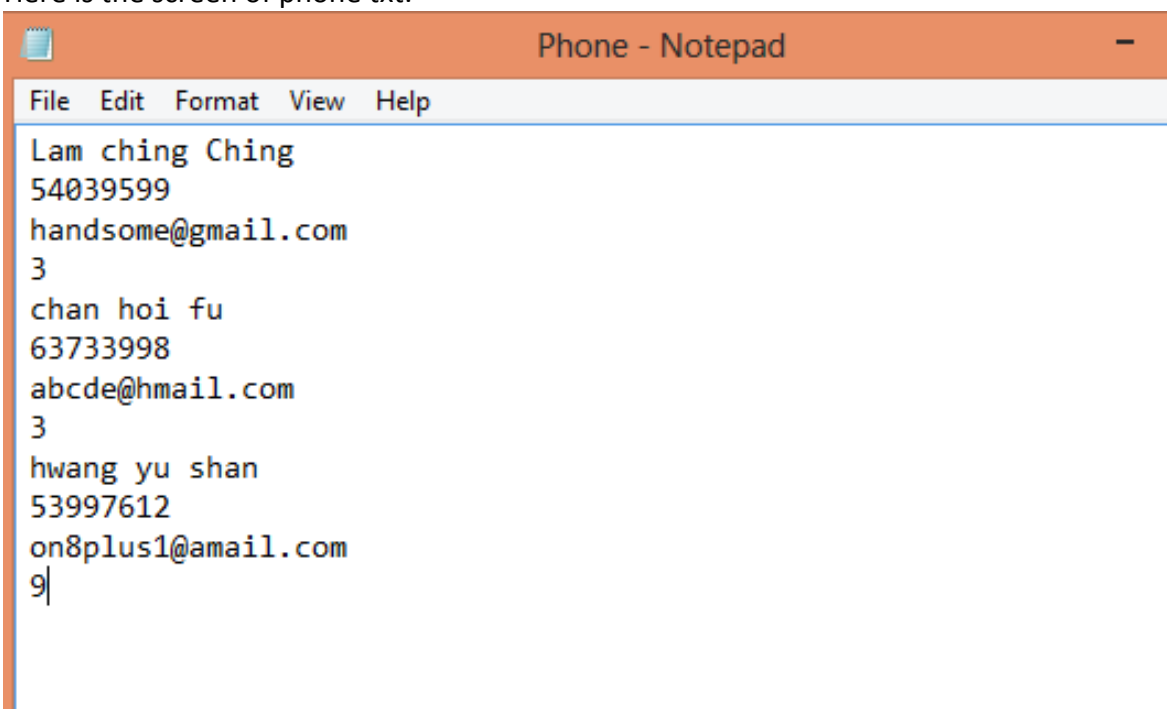
3) Upcase

Example:

```
 if upcase(ContactName[i]) = upcase(FindContactName) then
```

Users can type both up- case or lower-case as this function can let all input become up-case letter.

Here is the screen of phone txt.



3) Case

Example: case Choice of

1 : DisplayData;

2 : ChangeData;

3 : DeleteData;

4 : AddData

The case function will get the character which the users inputted. If the character is match with one of the case, it will do that case. Such as the character is 1 in the above example, the program will go to procedure display elective.

3.3 Procedure in the program

●Procedure display data

It is the procedure used to display the data stored in phone book

●Procedure change data

It is the procedure used for changing the data stored in phone book

●Procedure delete data

It is the procedure used to delete the data stored in phone book

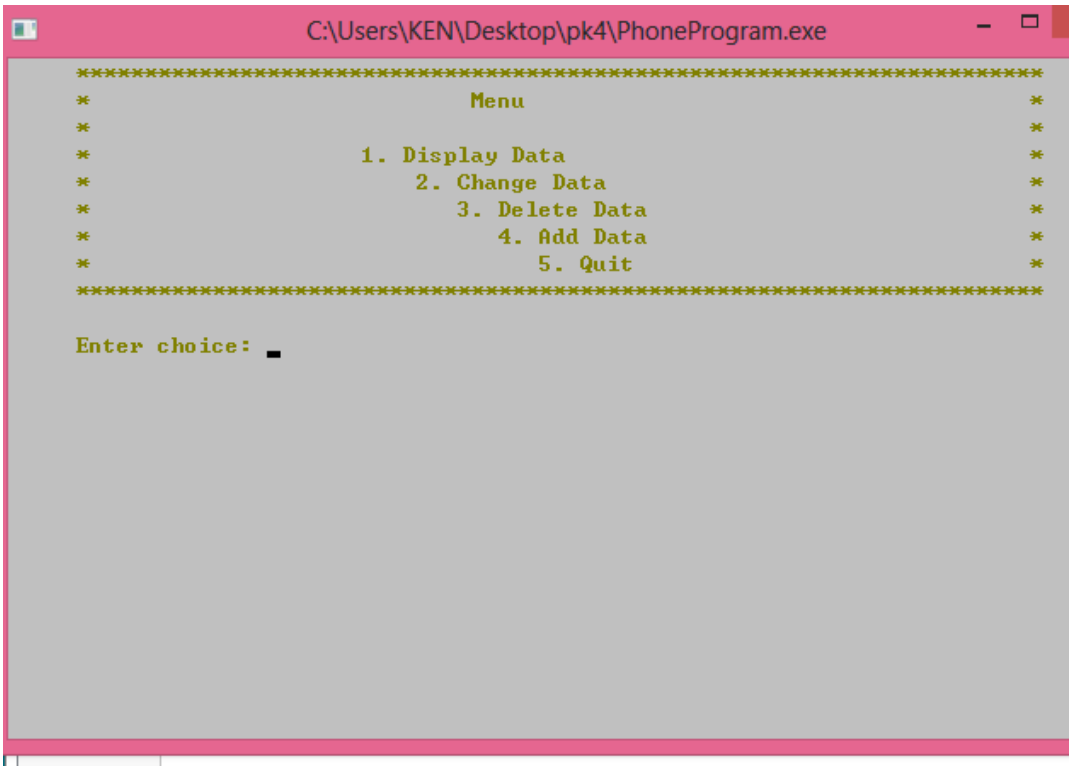
●Procedure add data

It is the procedure used to add new data to phone book

3.4 Program execution

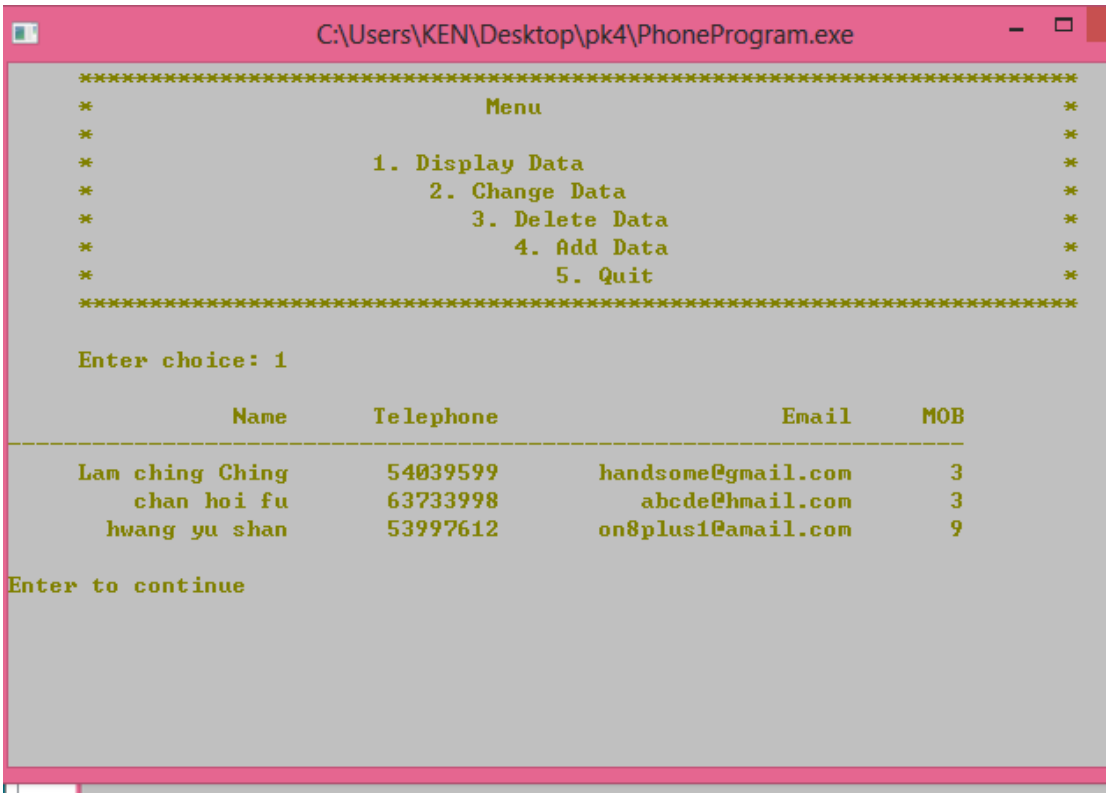
All the text files are placed at the same folder to make the program run correctly.

●Main menu



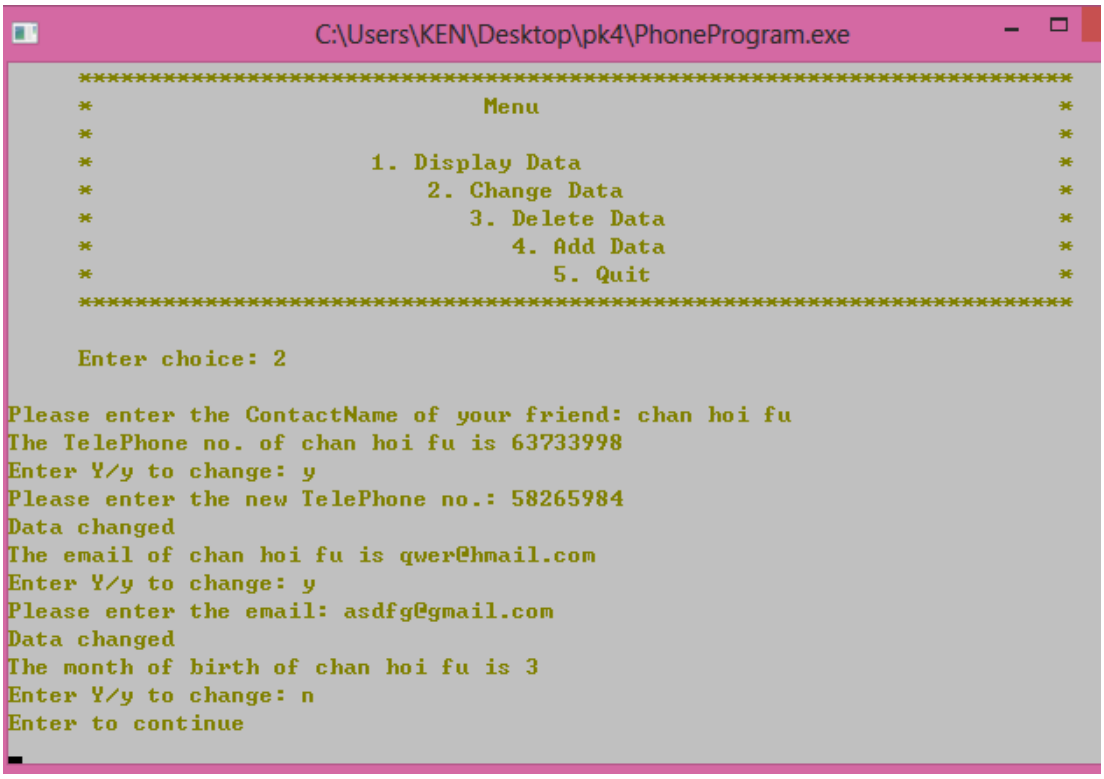
The above page is the main menu of the program. Users will first go into this Page when they execute the program. In this page, users have five choice which are display data , change data , delete data , add data and quit. Users can choose by entering numbers(1,2,3,4,5)

●**Display data**



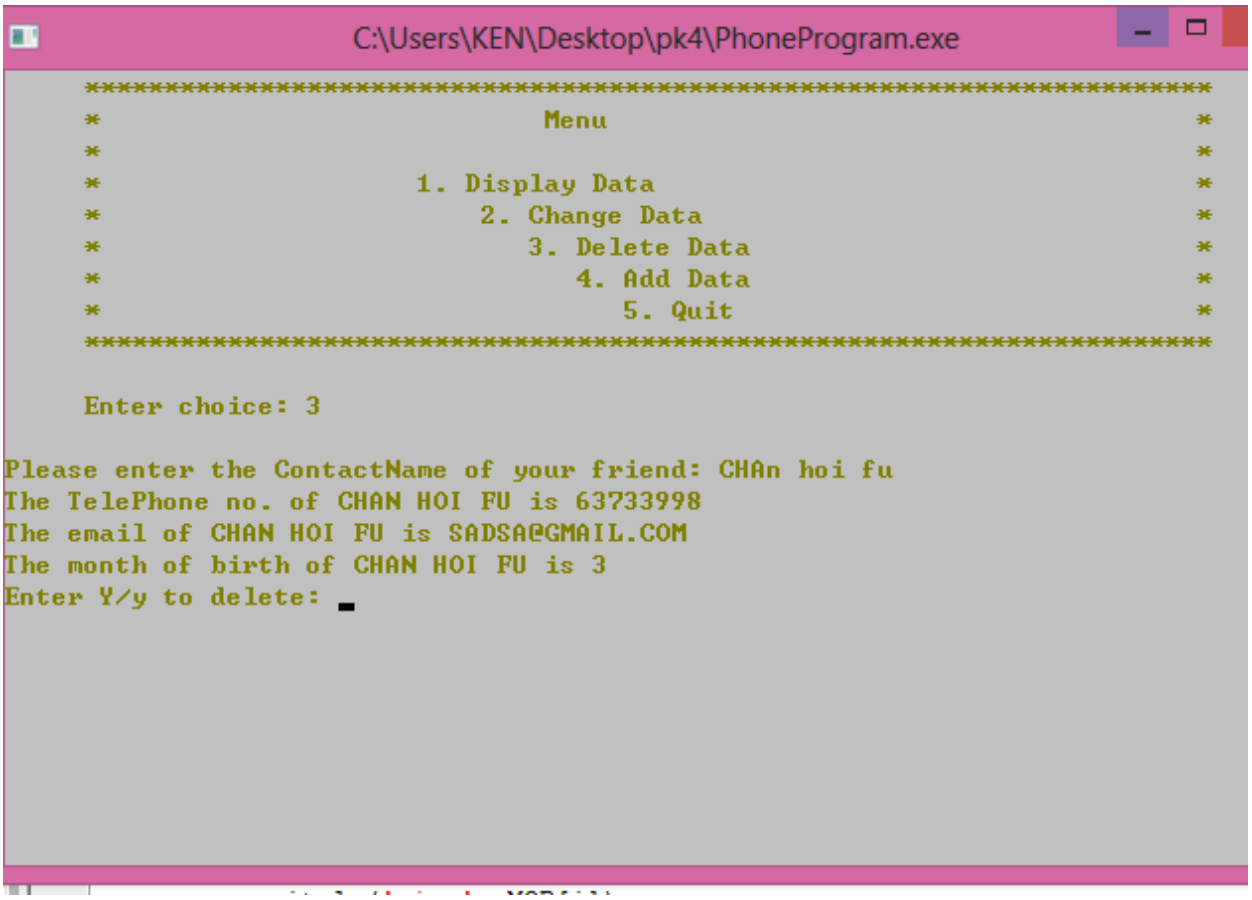
The above page is data display page. Users can go into this page by entering 1. This page shows the data which contains name, telephone, email and month of birth that users stored before. It is the page for users to check whether they need to correct their program or not. “Enter to continue” will be shown in order to remain users

●**Change data**



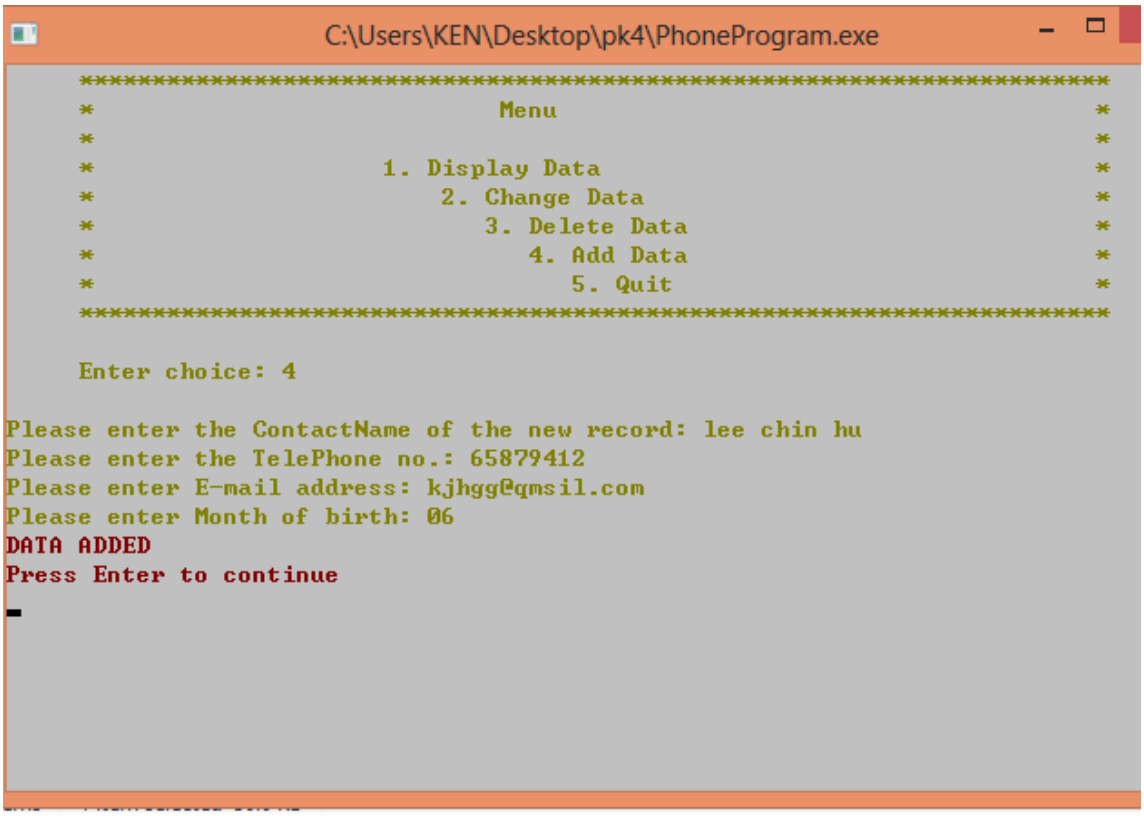
The above page is change data page. Users can change the telephone, email and month of birth that all stored before. Users can enter y/Y for changing data or any other characters for remaining. After changing data, “Enter to continue” will be shown in order to remain users

●**Delete data**



The above page is delete data page. Users can delete the unwanted, old telephone, email or month of birth that they stored before. Users can enter y/Y for changing or any other characters for remaining.

●**ADD data**



The above page is add data page. Users can add new data include telephone, email or month of birth into their phonebook. After adding new data, “DATA ADDED” AND “Press Enter to continue” will be shown in order to remain users

Chapter 4 Testing and Evaluation

4.1 brief description

As every program is not perfect at all, testing and evaluation is a must for all program. There will be some loopholes and bugs in the program such as logical or run-time error will be also included. With this reason, this part is used to check can the program run properly and meet the aim.

To begin with, I have to check whether the program can meet all the structure from the design part. I have to look at the function which in design part and test can the program can achieve the mission and has all these function.

After that, as one of the main reasons in this chapter is to make the program more user-friendly. Some debug works are important to make the program become more user-friendly. So debugging will be the next step after testing.

Besides, although some debugging had been done by me, there may still contain bugs that I cannot find out at all, so what I am going to do is let others to try and use my program. So that bugs may find out while using.

Last but not least, I will modify the program to make it perfect for users to use.

4.2Testing and Evaluation Plan

The program will be tested and evaluated under the plan below.
I will test the program firs tat all. It is important for me to test it first because there may be bugs that users cannot discover easily. For all the functions, at least five times testing should have been done. As there will be a handful of possible bugs, both correct and incorrect data should be tried to ensure the accuracy of the program. After finding all the bugs, debug is required but at the same time.

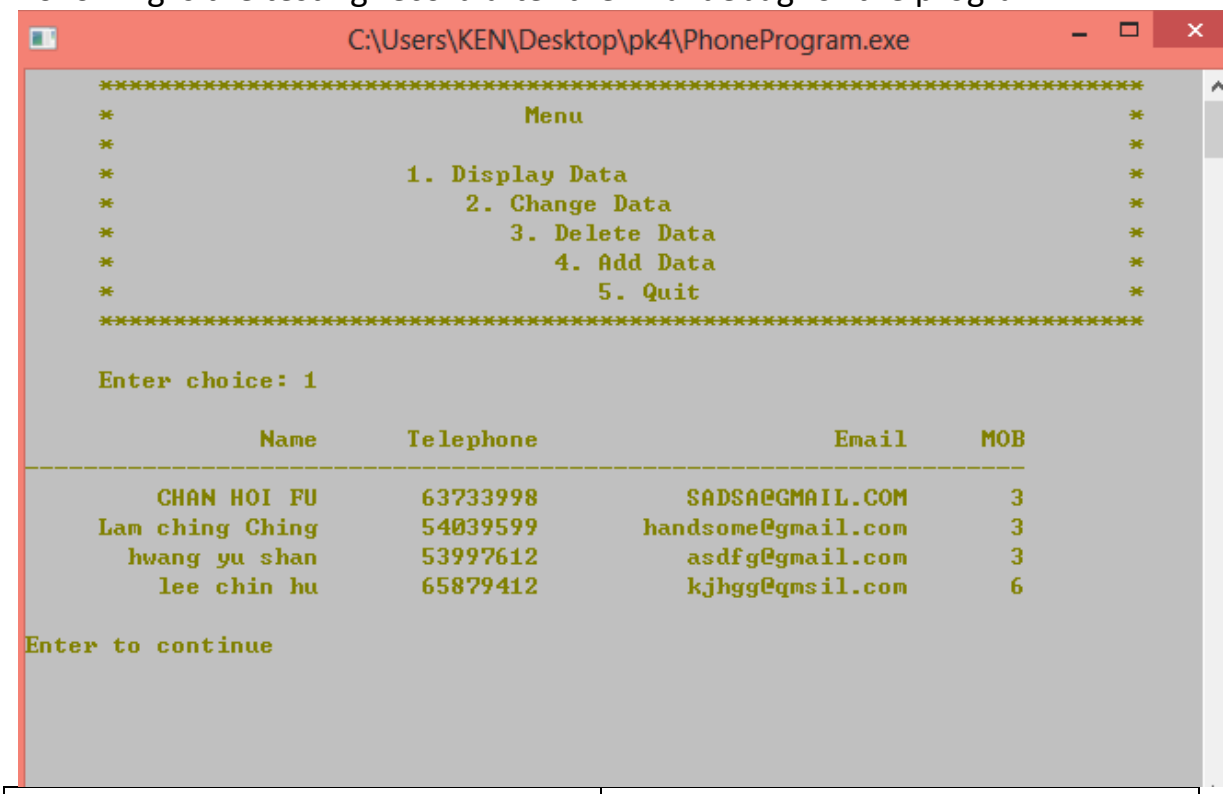
My own test will be focus on logic and run-time error. Since all the syntax error were debugged after finished the program. I have to check logic and run-time error this time. Run-time error is quite brothering. Since it is harder to debug, as Pascal do not warn you where is the error. As for logic error, it is the hardest error for debugging. Since neither Pascal nor the program will show errors to you. Only if enter the testing data can find the errors. So I will focus on these two errors

However, in view of a programmer, I may not find out all the error as people will be subjective on works done by them. So, I will let5 people who do not have much knowledge in ICT and 5people who have studied ICT to try my program and see whether any bugs can be exist.

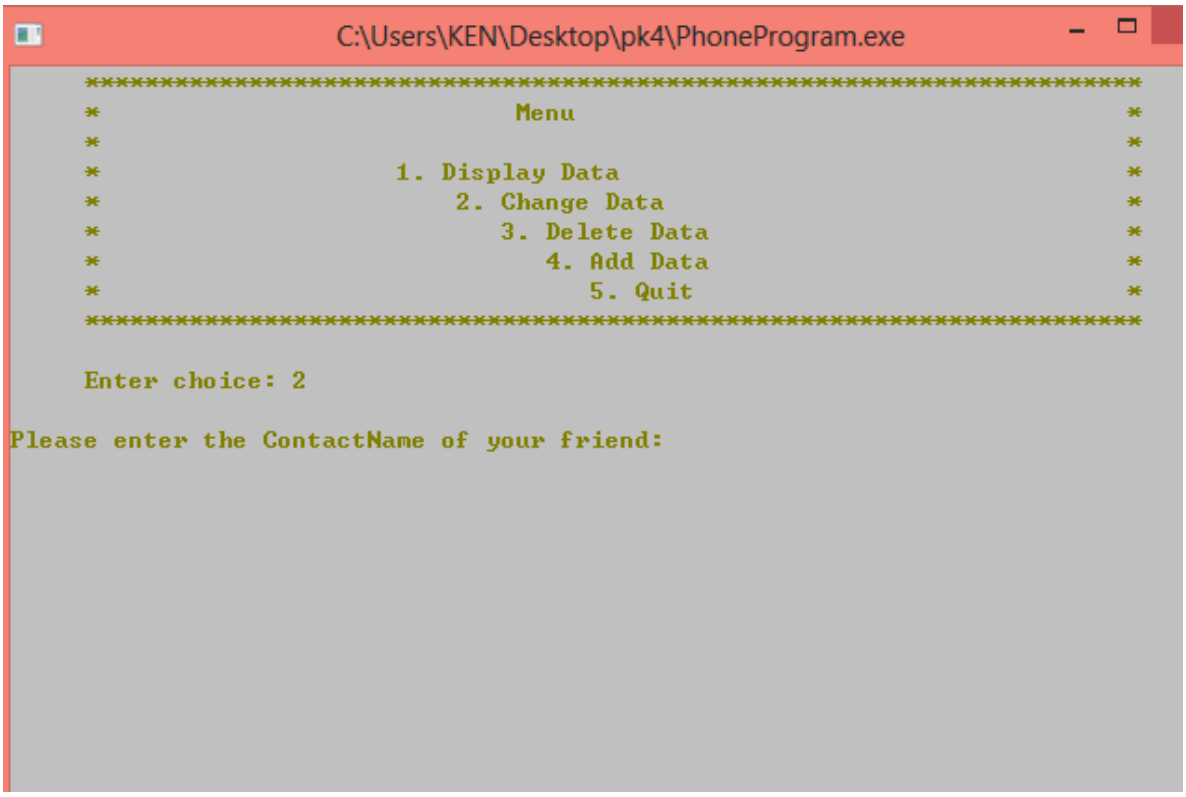
I will try to modify the program according to the reported bugs and suggestions. The new version of program after making modifications will be shown in Appendix 1.

4.3 Internal Testing

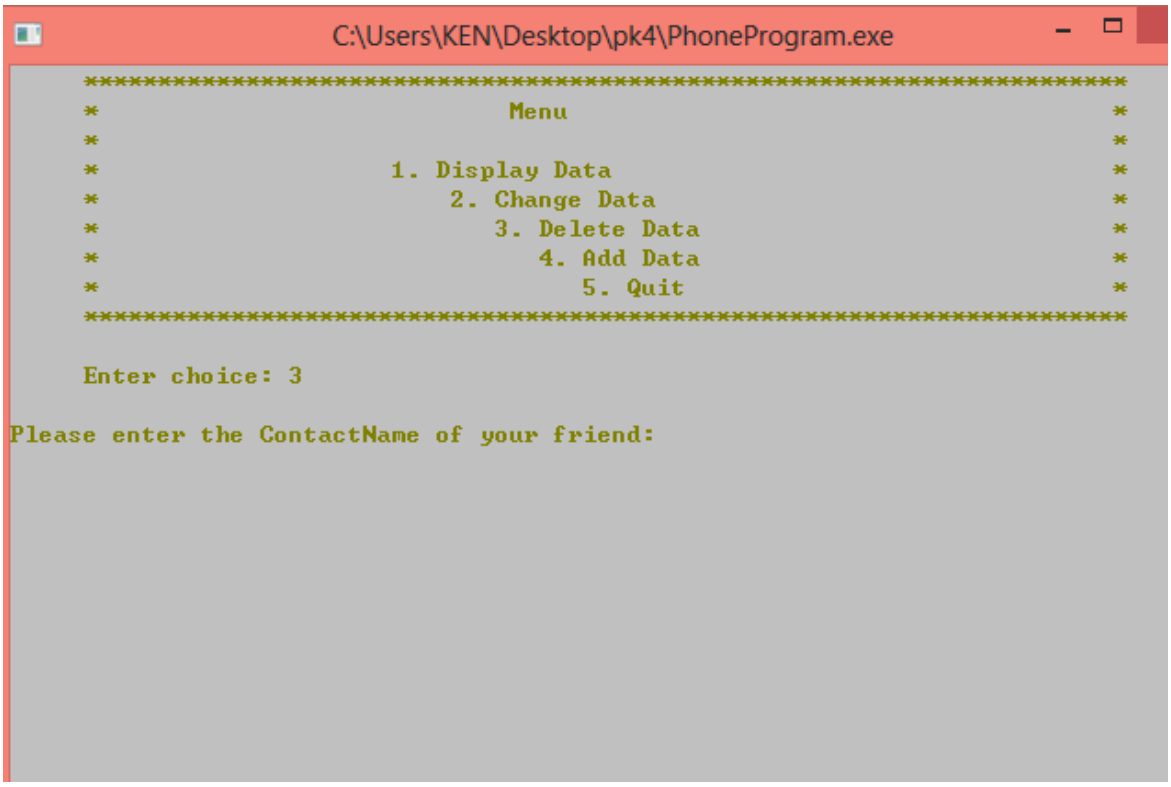
Following is the testing record after the final debug for the program :



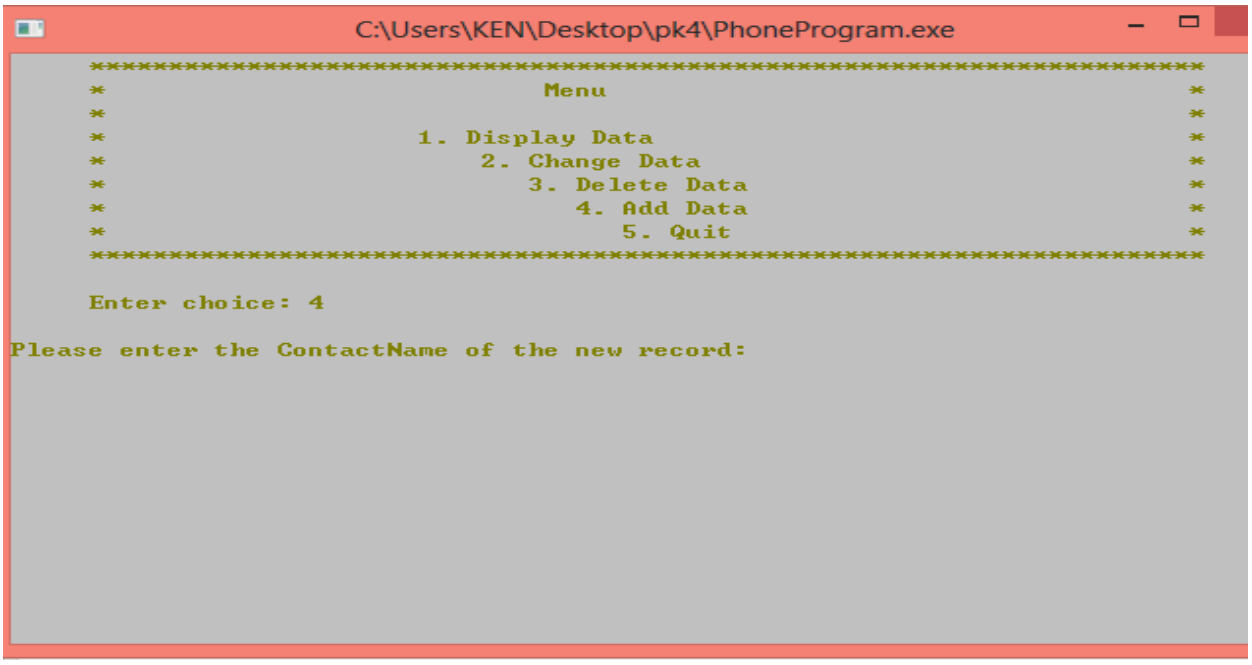
Purpose:	To check whether enter'1' can go into display data page
Input:	'1'
Expected Output:	Go into display data page
Actual Output:	Go into display data page
Test Result:	Pass/No bugs found
Follow-up action:	Nil



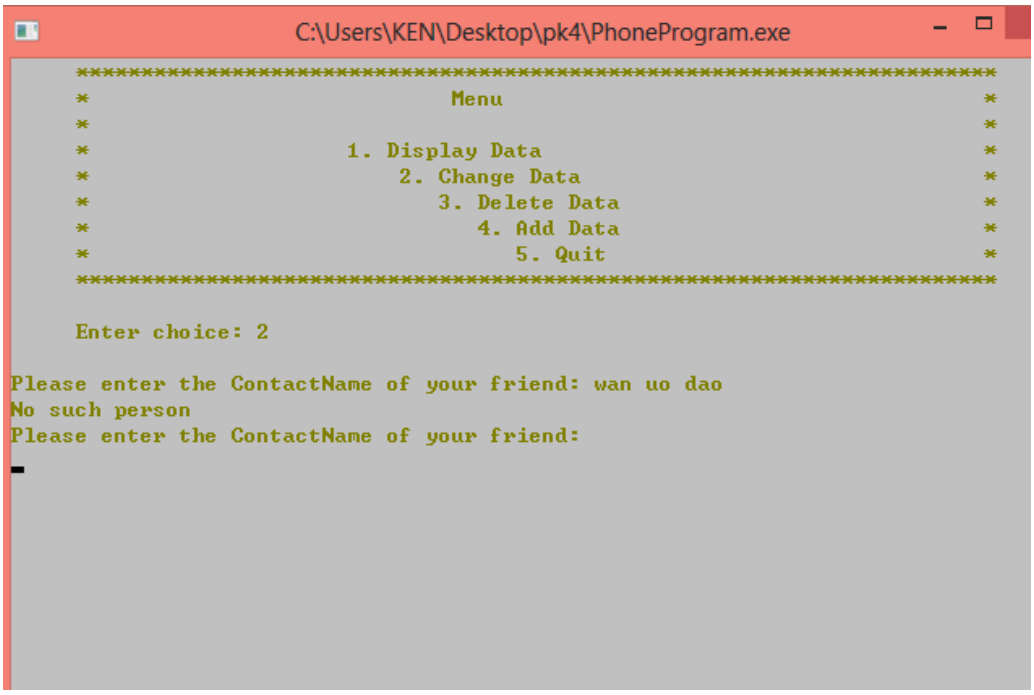
Purpose:	To check whether enter'2' can go into change data page
Input:	'2'
Expected Output:	Go into change data page
Actual Output:	Go into change data page
Test Result:	Pass/No bugs found
Follow-up action:	Nil



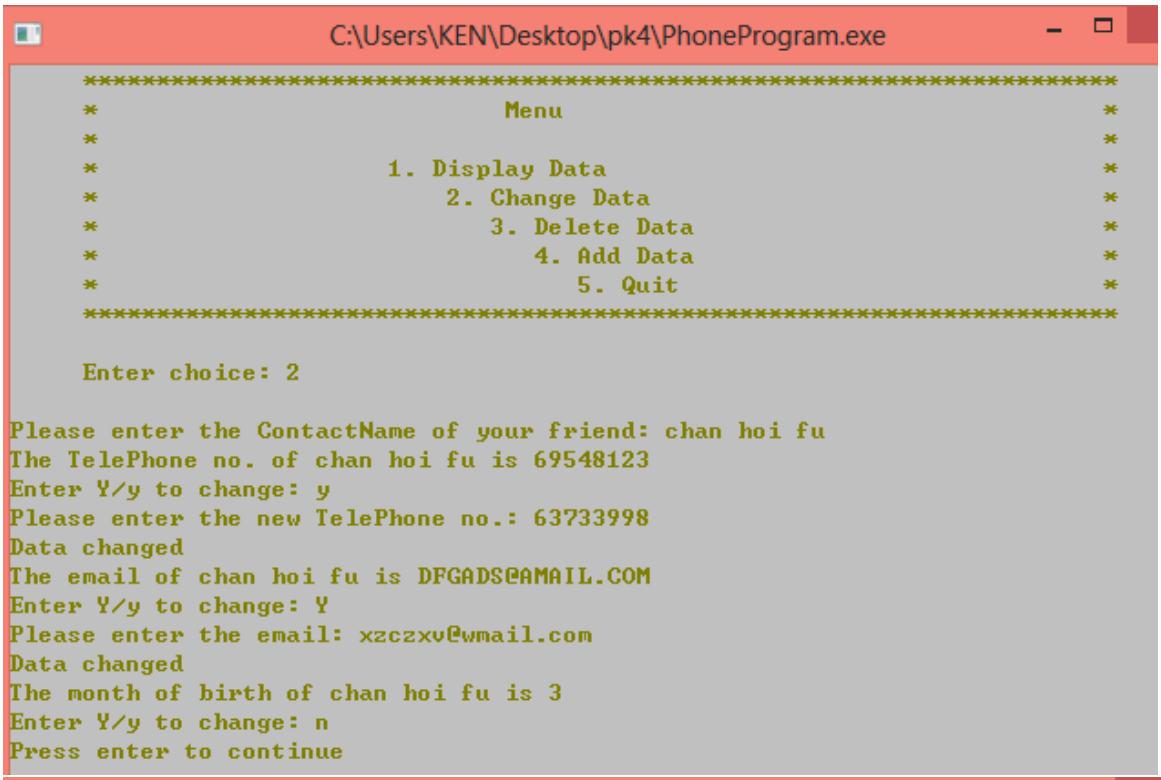
Purpose:	To check whether enter'3' can go into delete data page
Input:	'3'
Expected Output:	Go into delete data page
Actual Output:	Go into delete data page
Test Result:	Pass/No bugs found
Follow-up action:	Nil

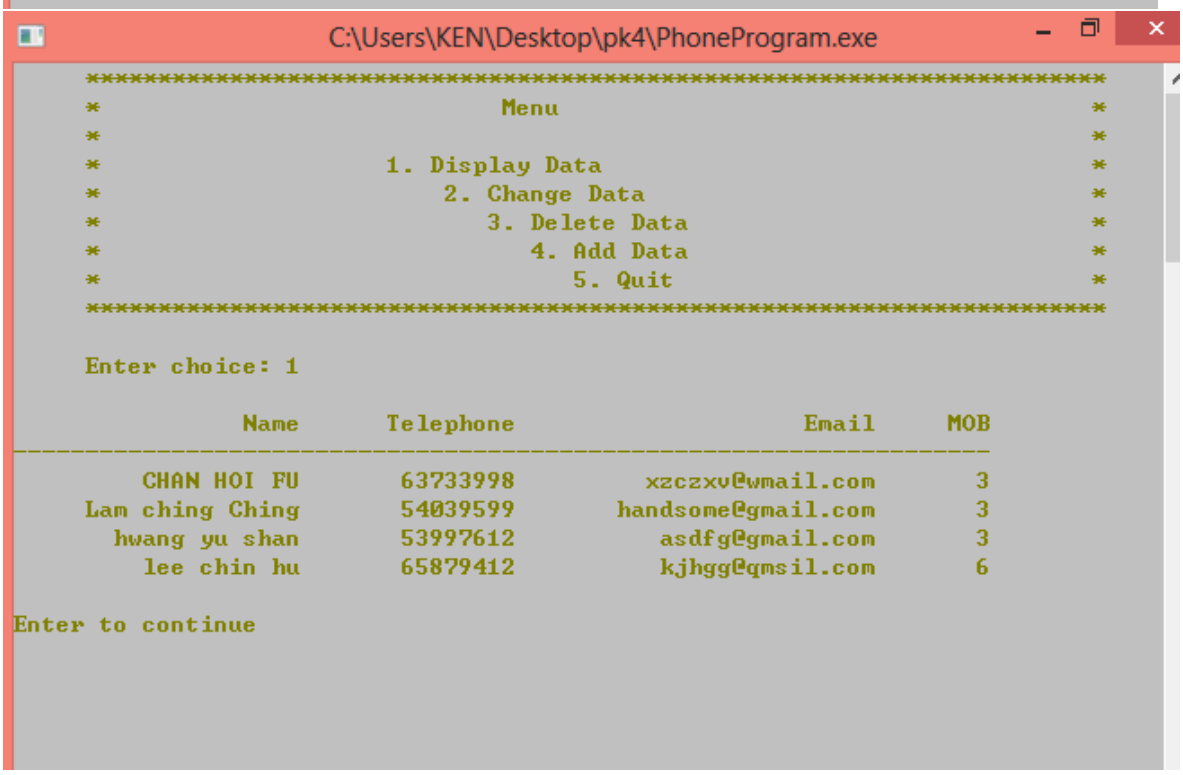


Purpose:	To check whether enter'4' can go into add data page
Input:	'4'
Expected Output:	Go into add data page
Actual Output:	Go into add data page
Test Result:	Pass/No bugs found
Follow-up action:	Nil



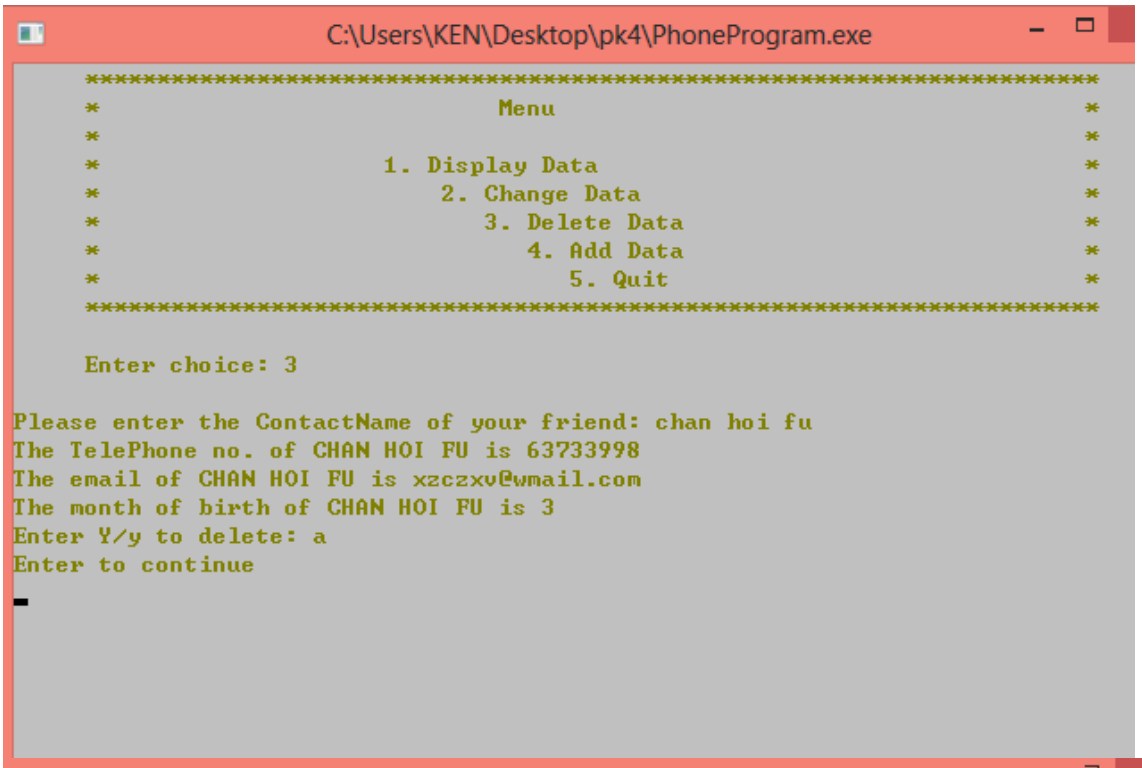
Purpose:	To check whether the remainder “No such person”will be shown when enter the data that did not exist in stored data.
Input:	The data that did not exist in stored data
Expected Output:	“No such person” will be shown and enter again
Actual Output:	AS same as expectation
Test Result:	Pass/No bugs found
Follow-up action:	Nil

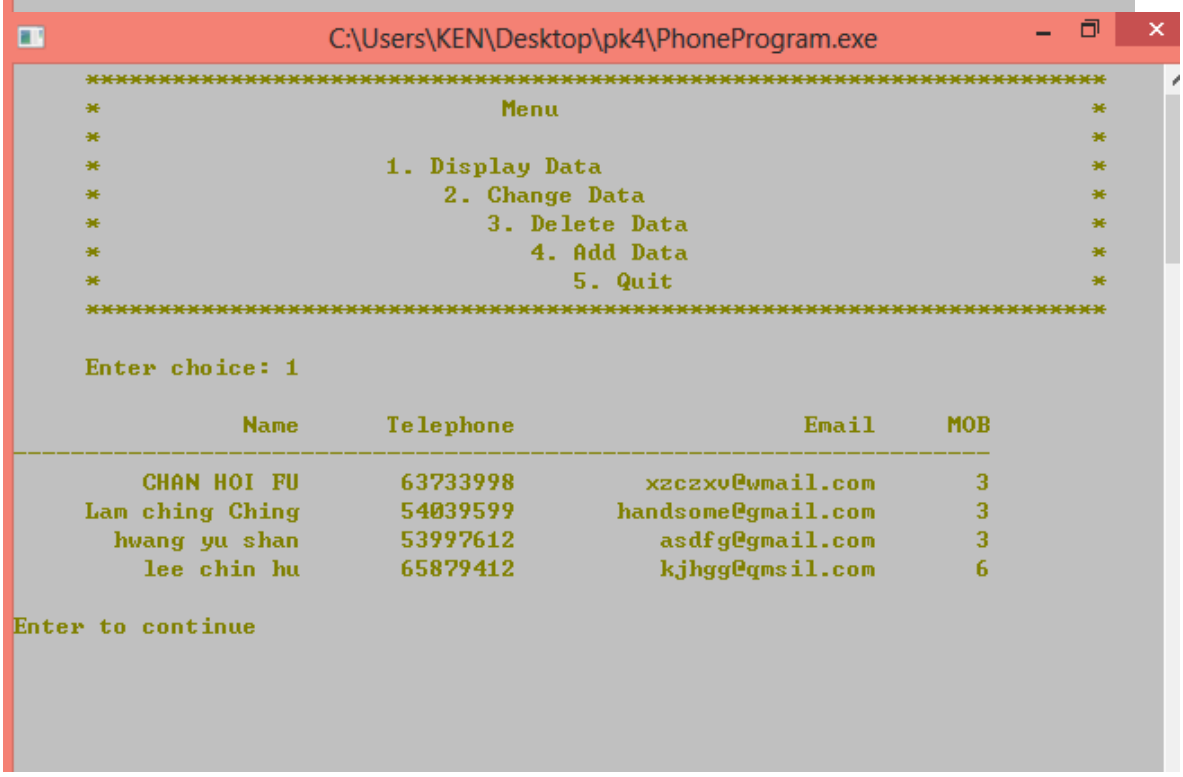


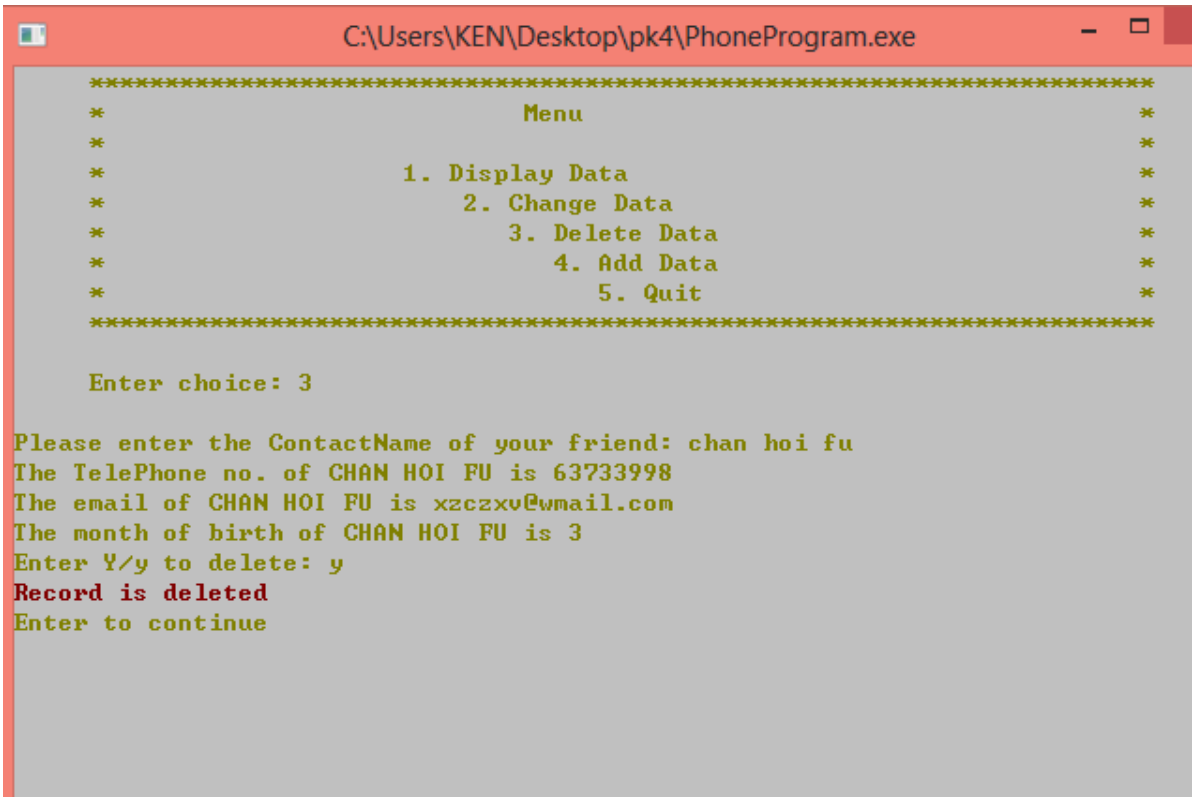


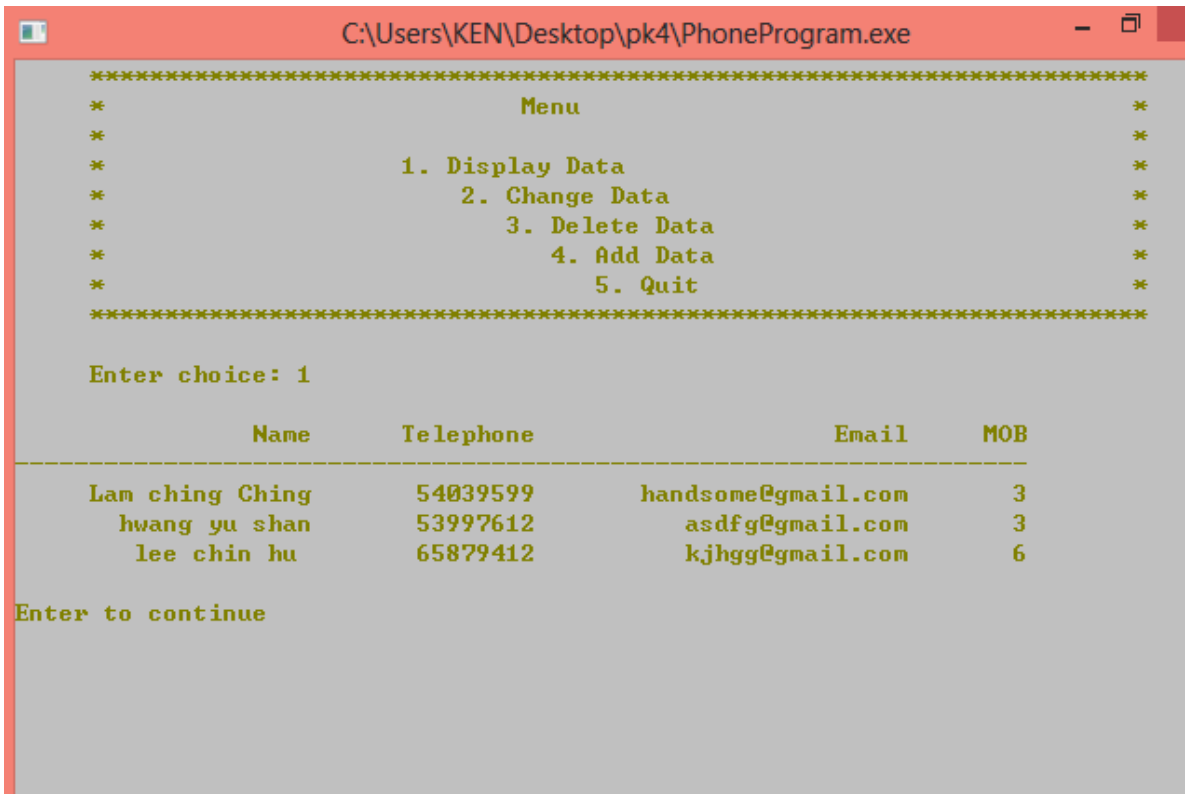
Purpose:	To check whether enter y/Y will show the next step that require use to enter the data and other characters will remain the data unchanged
Input:	Y, y and n(other characters)
Expected Output:	1. Input Y or y will show the next step that require use to enter the data 2. Input n(other characters) will remain the data unchanged

Actual Output:	AS same as expectation
Test Result:	Pass/No bugs found
Follow-up action:	Nil


C:\Users\KEN\Desktop\pk4\PhoneProgram.exe
****
* Menu *
*
* 1. Display Data *
* 2. Change Data *
* 3. Delete Data *
* 4. Add Data *
* 5. Quit *
****
Enter choice: 3
Please enter the ContactName of your friend: chan hoi fu
The TelePhone no. of CHAN HOI FU is 63733998
The email of CHAN HOI FU is xzczxv@wmail.com
The month of birth of CHAN HOI FU is 3
Enter Y/y to delete: a
Enter to continue

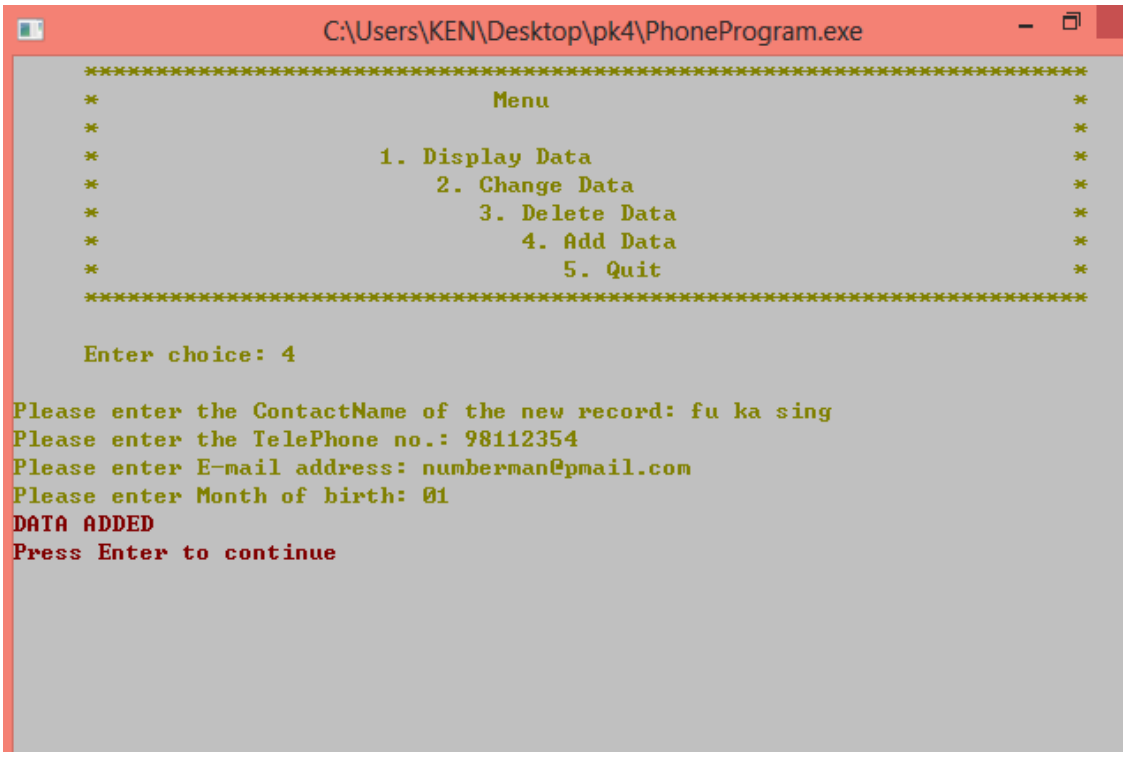

C:\Users\KEN\Desktop\pk4\PhoneProgram.exe
****
* Menu *
*
* 1. Display Data *
* 2. Change Data *
* 3. Delete Data *
* 4. Add Data *
* 5. Quit *
****
Enter choice: 1
Name Telephone Email MOB
CHAN HOI FU 63733998 xzczxv@wmail.com 3
Lam ching Ching 54039599 handsome@gmail.com 3
hwang yu shan 53997612 asdfg@gmail.com 3
lee chin hu 65879412 kjhgg@qmsil.com 6
Enter to continue



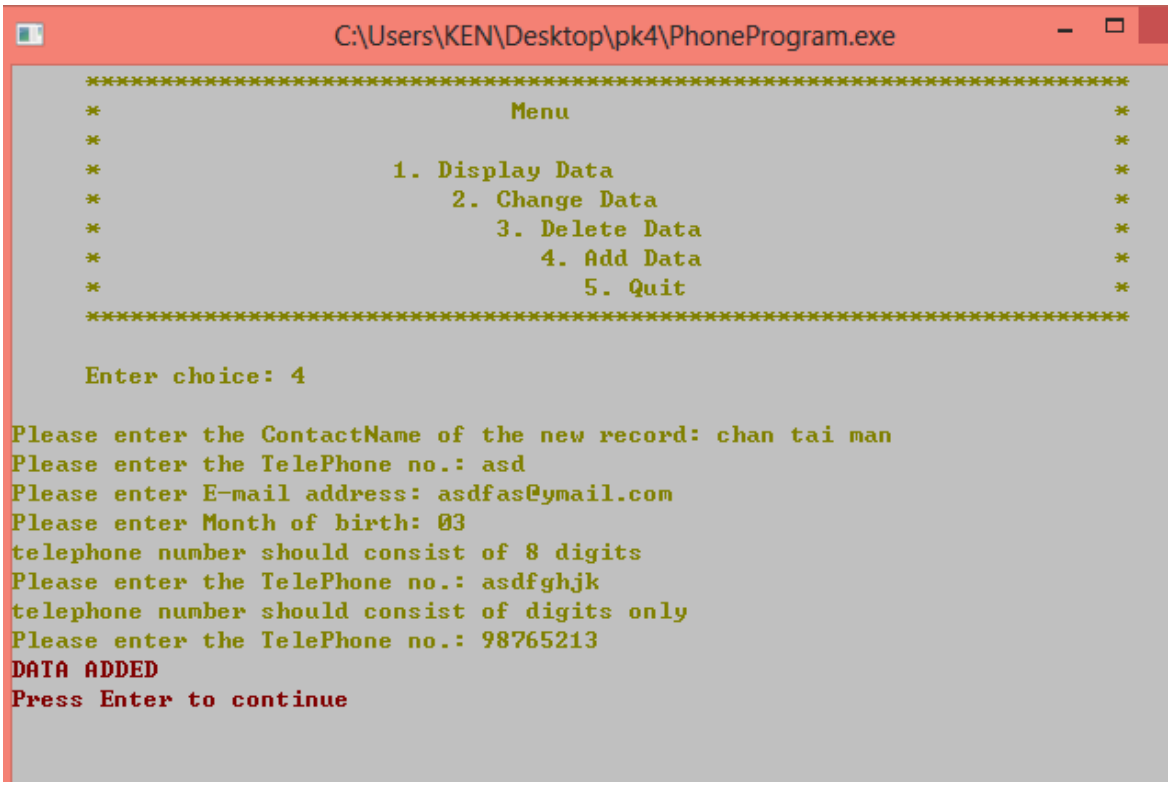


Purpose:	To check whether enter y/Y will delete the data and other characters(a) will remain the data
Input:	Y/y and a(other characters)
Expected Output:	1.Y/y will delete the data 2.other character will remain the data unchanged

Actual Output:	AS same as expectation
Test Result:	Pass/No bugs found
Follow-up action:	Nil



Purpose:	To check whether contact name, telephone, email and month of birth can be added . Also , show the remainder “ DATA ADDED” and ‘ Press Enter to continue”.
Input:	contact name, telephone, email and month of birth
Expected Output:	“ DATA ADDED” and ‘ Press Enter to continue”.
Actual Output:	AS same as expectation
Test Result:	Pass/No bugs found
Follow-up action:	Nil



Purpose:	To check whether the remainders are work or not
Input:	1. incorrect type of phone number(English word) 2. non-8digits input
Expected Output:	"Telephone number should consist of 8 digits" and " Telephone should consists of digits only"
Actual Output:	AS same as expectation
Test Result:	Pass/No bugs found
Follow-up action:	Nil

4.4External Testing and Evaluation

15 of my classmates have been invited to use my program so that while they are playing on it, they may discover bugs and loopholes in my program which I cannot find out in the previous sections.

A simple testing form is made to make them clear on what they should do when they are playing on the program. In here, I want to thank all people who are helping me on finding out the problems in my program.

After collecting all the data and opinion from the classmates, I have summarized all of them and do improvement.

Thank you for helping me to test the program!

Please circle following number.

1.	The interface is well-designed	5	4	3	2	1
2.	The function is useful	5	4	3	2	1
3.	The program is user-friendly	5	4	3	2	1
4.	The idea is good	5	4	3	2	1

5 is the best , 1 is the worst

It there any bugs or improvement?

Which part of program you like the most?

Other suggestions

After collecting the data, the result is shown below

1.	The interface is well-designed	4.6
2.	The function is useful	4.8
3.	The program is user-friendly	4.5
4.	The idea is good	4

From the above result, I know that lots of interviewees are satisfied with program's function and interface

However, the rate of idea is relatively low. It is because phonebook is very common and I should add more field for people to store other useful things Like facebook name.

Chapter 5 Conclusion & Discussion

Pros and Shortcoming of my Program

During this report, I have paid a lot of effort in order to make the program more perfect especially in planning, designing, implementing. However, It is hard to make the program zero mistake. I believe that I have gave my best to do the program.

My program is to design a phonebook. The interface of the program is well-designed with a colourful background and I have indicate all the instruction clearly , so that user can follow the remainder when they encounter some problem

The look is only the surface of my program, but more importantly is that the content of the program is also wonderful.

And I think that the most important feature of my program is users can add more useful data such as email and month of birth in the phonebook as people can just store name and telephone number in past

Unfortunately, the program still has shortcomings since every program are not perfect. The information of the user that display on the program may not enough and the interface can be done better by adding new element like flash.

But consider all the system, the program can work properly and smoothly which can make the users to use it with a high efficiency.

Finally, I have confidence that the program will be useful for the users

Future Improvement

Although this phonebook is good enough for saving the contact details like name , telephone number , email and month of birth , I think many data such as facebook name field can be added for users to stored .However , the larger the program are , the easier the loophole exist. Therefore, I prioritize email, month of birth for users to stored.

Moreover, the interface can be design better. In this phonebook, I use colorful background and bubble to design the interface. Also, the remainder sentences is use different color for users to distinguish. However, i think there are still many

Elements such as flash, reverse that I can added to the program after I acquire more computer knowledge.

Self reflection

This is the first time that I do this kind of computer program and reports. It is not easy at the beginning as I am not familiarize with the format. Lucky, all my friends

And teachers are willing to give a helping hand to me when I encounter technical problems. At first, I want to design a subject allocation system as I think it is more

Easy to do. However, I found that it is far more difficult as I think in the research period. So I choose to design phonebook that is more close to life and common.

Once I learn to design phonebook, I can apply to it in daily life.

Chapter 6 Reference and Acknowledgement

Reference

-Internet resources

Hong Kong computering forum

-Teacher supervisor

Mr.Chu

Appendix 1: Program Code

```
program PhoneBook;



Uses Crt;


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


var  ContactName : array[1..Max] of string;
     TelePhone : array[1..Max] of string;
     Email : array[1..Max] of string;
     MOB : array[1..Max] of integer;
     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, ContactName[i]);
      readln(Infile, TelePhone[i]);
      readln(Infile, Email[i]);
      readln(Infile, MOB[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, ContactName[i]);
      writeln(Outfile, TelePhone[i]);
      writeln(Outfile, Email[i]);
      writeln(Outfile, MOB[i])
    end;
  close(Outfile)
end;

procedure DisplayData;
var i : integer;
begin

  writeln('              Name      Telephone                Email     MOB');
  writeln('-------------------------------------------------------------------');
  for i := 1 to N do
    begin
      writeln(ContactName[i]:20,TelePhone[i]:15,Email[i]:25,MOB[i]:8);
    end;
  writeln;

  write('Enter to continue');
```

```
  readln
end;



procedure ChangeData;
var i : integer;
    FindContactName : string;
    Choice: char ;
    Found:Boolean;
begin
  write('Please enter the ContactName of your friend: ');
  readln(FindContactName);
  Found:=false;
  for i := 1 to N do
    if upcase(ContactName[i]) = upcase(FindContactName) then
     begin
      Found:=true;
            write('The TelePhone no. of ');
            write(FindContactName);
            writeln(' is ', TelePhone[i]);
      write('Enter Y/y to change: ');
      readln(choice);
      If (choice = 'Y') OR (choice = 'y')
      then
```

```
  begin
    write('Please enter the new TelePhone no.: ');
          readln(TelePhone[i]);
  writeln('Data changed');



   end;
write('The email of ');
write(FindContactName);
writeln(' is ', email[i]);
write('Enter Y/y to change: ');
readln(choice);
If (choice = 'Y') OR (choice = 'y')
then
  begin
    write('Please enter the email: ');
          readln(email[i]);
  writeln('Data changed');
  end ;
  write('The month of birth of ');
write(FindContactName);
writeln(' is ', MOB[i]);
write('Enter Y/y to change: ');
readln(choice);
```

```
    If (choice = 'Y') OR (choice = 'y')
    then
      begin
        write('Please enter the month of birth: ');
              readln(MOB[i]);



      end
    end;

    TextCOLOR(6);
  if Found=false then
    writeln('No such person');
    WRITELN('Enter to continue');
    TextCOLOR(6);
  WriteFile
end;




procedure DeleteData;
var i,j, Locate : integer;
```

```
    FindContactName : string;
    choice : char;
begin
  write('Please enter the ContactName of your friend: ');
  readln(FindContactName);
  Locate := 0;
  for i := 1 to N do
    if upcase(ContactName[i]) = upcase(FindContactName)
      then
        begin
          write('The TelePhone no. of ');
          write(ContactName[i]);
          writeln(' is ', TelePhone[i]);
           write('The email of ');
          write(ContactName[i]);
          writeln(' is ', email[i]);
           write('The month of birth of ');
          write(ContactName[i]);
          writeln(' is ', MOB[i]);
          Locate := i;
          write('Enter Y/y to delete: ');
          readln(choice);
          If (choice = 'Y') OR (choice = 'y')
           then
```

```
          begin
           for j := Locate to N-1 do
                  begin
              ContactName[i] := ContactName[i+1];
                   TelePhone[i] := TelePhone[i+1]
                  end;
           TextCOLOR(red);
                 writeln('Record is deleted');

           TextCOLOR(6);
           N := N - 1
          end
       end;
       TextCOLOR(red);
  if Locate = 0 then

    writeln('No such person');

    TextCOLOR(6);
     WRITELN('Enter to continue');
  WriteFile
end;

procedure Adddata;
```

```
var NewContactName,
   NewTelePhone,NewEmail : string;
   NewMOB,i, P: integer;
   vaildphone:boolean;
begin
  write('Please enter the ContactName of the new record: ');
  readln(NewContactName);
  write('Please enter the TelePhone no.: ');
  readln(NewTelePhone);
  write('Please enter E-mail address: ');
  readln(NewEmail);
  write('Please enter Month of birth: ');
  readln(NewMOB);

  repeat
   vaildphone:= true;
   if length(NewTelePhone)<>8 then
    begin
     vaildphone:= false;
     writeln('telephone number should consist of 8 digits');
     write('Please enter the TelePhone no.: ');
     readln(NewTelePhone);
    end
   else
```

```
  for i := 1 to 8 do
    if (NewTelePhone[i] < '0') or (NewTelePhone[i] > '9')
    then begin
      vaildphone:= false;
      writeln('telephone number should consist of digits only');
      write('Please enter the TelePhone no.: ');
      readln(NewTelePhone);
    end
until vaildphone ;
TextCOLOR(red);
WRITEln('DATA ADDED');
writeln('Press Enter to continue');
 TextCOLOR(6);



P := N + 1;
for i := N downto 1 do
  if ContactName[i] > NewContactName
    then P := P - 1;
for i := N downto P do
  begin
    ContactName[i+1] := ContactName[i];
    TelePhone[i+1] := TelePhone[i];
```

```
    Email[i+1] := Email[i];
    MOB[i+1] := MOB[i];
   end;
  ContactName[P] := NewContactName;
  TelePhone[P] := NewTelePhone;
  Email[P] := NewEmail;
  MOB[P] := NewMOB;
  N := N + 1;
  WriteFile
end;


begin  { Main Program }

  ReadFile;
  Textbackground(WHITE);
  Textcolor(6);
  Repeat
   Clrscr;
   writeln('
**********************************************************************
****');
   writeln('    *                    Menu                          *' );
   writeln('    *                                                  *' );
   writeln('    *              1. Display Data                     *' );
```

```
   writeln('    *                  2. Change Data                    *' );
   writeln('    *                    3. Delete Data                  *' );
   writeln('    *                      4. Add Data                   *' );
   writeln('    *                        5. Quit                     *' );
   writeln('
**********************************************************************
****');
   writeln;
   write('     Enter choice: ');

   readln(Choice);
   writeln;
   case Choice of
     1 : DisplayData;
     2 : ChangeData;
     3 : DeleteData;
     4 : Adddata;
   end;
   readln
  until Choice = 5;

  WriteFile

end.
```

```
begin  { Main Program }


  ReadFile;
  Textbackground(WHITE);
  Textcolor(6);
  Repeat
    Clrscr;
    writeln('
**********************************************************************
****');
    writeln('    *                    Menu                          *' );
    writeln('    *                                                  *' );
    writeln('    *               1. Display Data                    *' );
    writeln('    *                  2. Change Data                  *' );
    writeln('    *                    3. Delete Data                *' );
    writeln('    *                     4. Add Data                  *' );
    writeln('    *                       5. Quit                    *' );
    writeln('
**********************************************************************
****');
    writeln;
    write('     Enter choice: ');


    readln(Choice);
    writeln;
```

```
    case Choice of
      1 : DisplayData;
      2 : ChangeData;
      3 : DeleteData;
      4 : Adddata;
    end;
    readln
  until Choice = 5;

  WriteFile

end.
```