Hong Kong Diploma of Secondary Education 201X

Information and Communication Technology

Course Work

Module D: Software Development

Title: Phone Book Directory Program

Contents

Chapter 1: Introduction

1.1 Background

Phone book directory is a necessity in our life, while the electronization of the world, electronic phone book can help people simply record the information of others and easily find out these informations by using the function of searching.

1.2 Objective

In this project, I am going to develop a phonebook directory program which having searching and sorting function, users can fill in their friends' name, phone, email address and some specific remark.

For the program, it is sample, so it is suitable for all people, no matter the child or an adult, users only need to follow the instructions of the interface, and they will successful manage their phonebook.

In the phonebook, there will not be a login function, the reason is sample, I aim that the user can manage and find their friends' phone directly and as soon as possible.

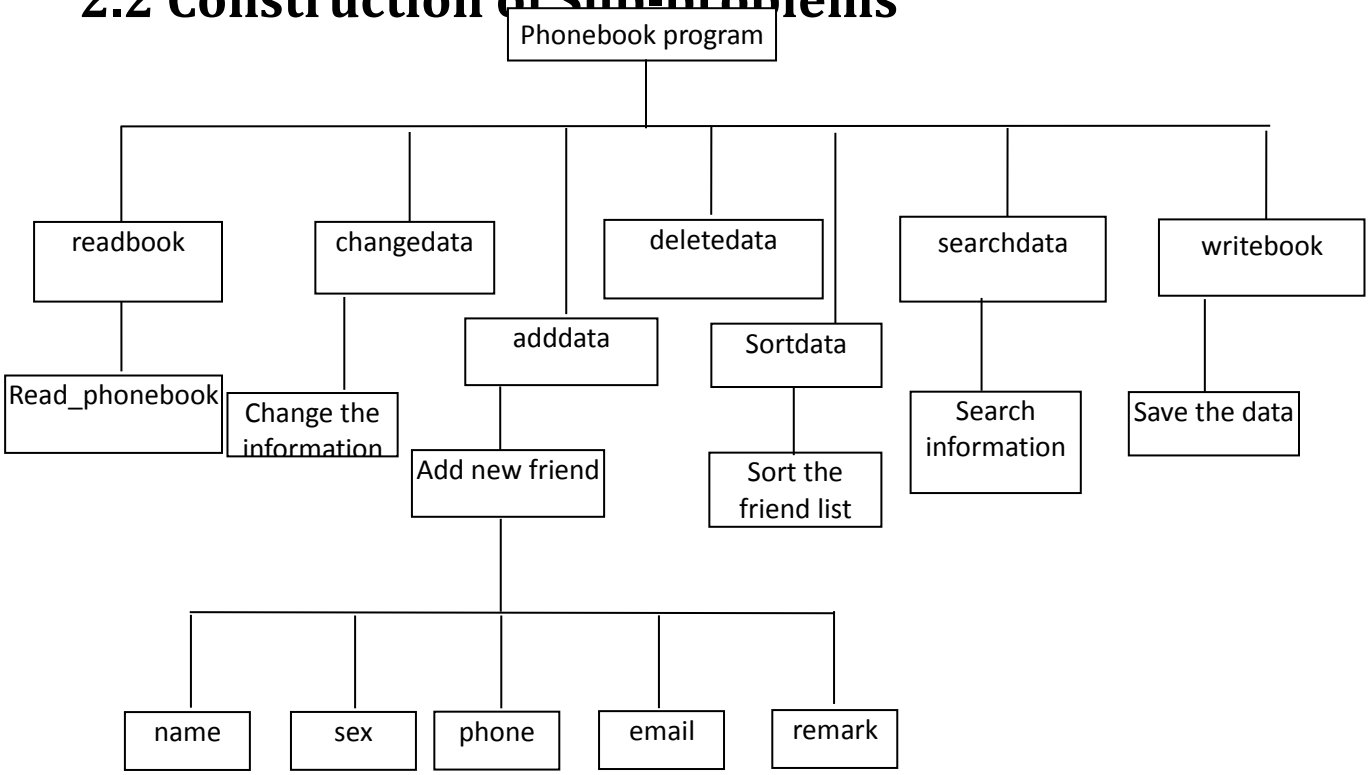
The sorting function in this program can help people who like tidily manage their phonebook, the phonebook content will be arranged alphabetically.

Chapter 2: Analysis

2.1 Proposed functions of the System

1. Storing email address
   - Important
   - Convenient for user to connect friends by other ways
2. Searching friends
   - Convenient for users to search their friends in a large friend list
   - Basic function
   - All the information will list out (name, sex, phone, email, remark)
3. Sorting the friend list
   - Convenient for users to tidy up the friend list
   - User-friendly

2.2 Construction of Sub-problems



Input data: name, sex, phone, email, remark

Process: Store in an array

Output: A sorted friend list

2.3 Choice of IT Tools

Software: Dev-Pascal

Reasons:

1. Pascal is simple, it is easy to use and learn.
2. Pascal has a fast run time.
3. Pascal uses a command-line user interface, which support more types of Windows.
4. No interpreter is needed.

OS-system: Windows

Reasons:

1. Windows is commonly used.
2. Provide a simple user interface.

Chapter 3: Design of Solution

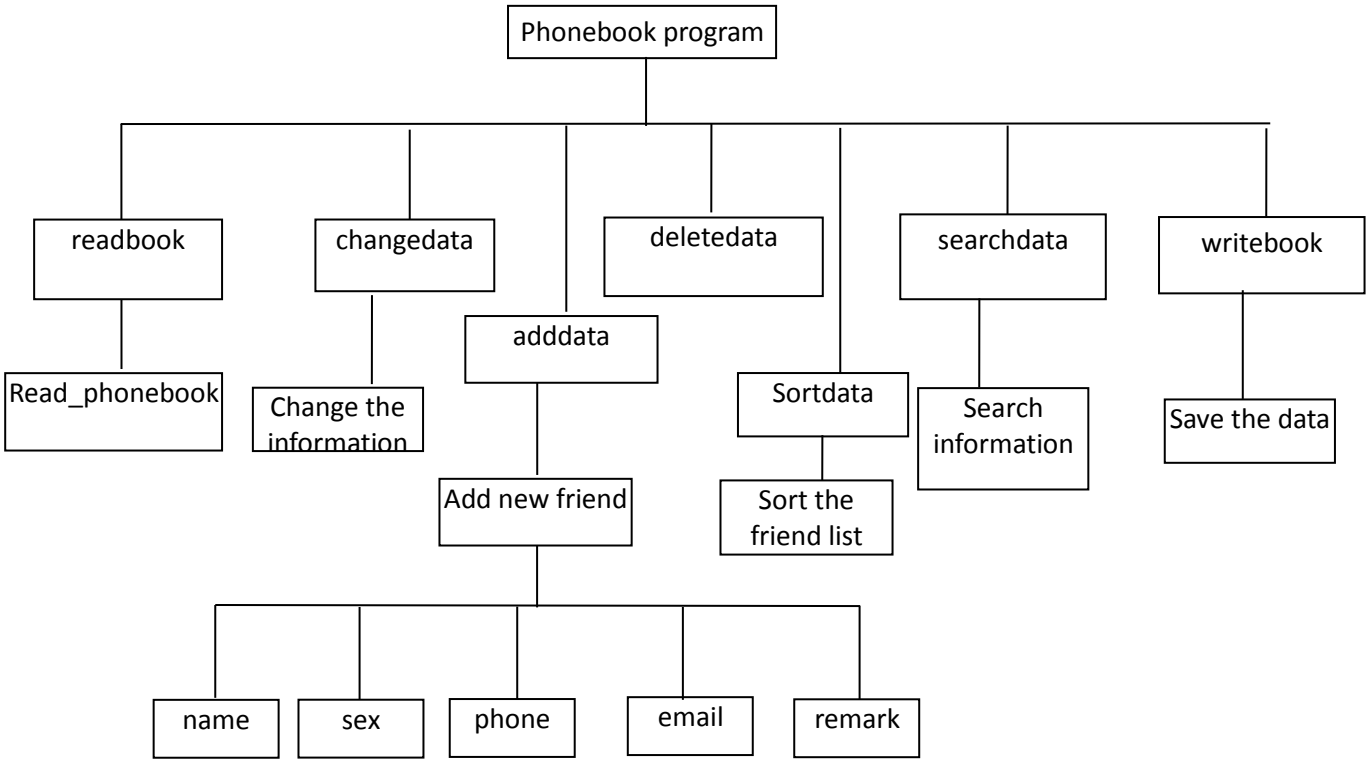
3.1 Brief Description

- To design the sub-programs (procedures).
- To design the main body of the program.

3.2 Design of the Main Program

- The main program is about the user interface at most.
- There is a menu for user.
- The main program provides choice for user to choose what they want to do.
- Choices for users:
    1. Insert the new friend
    2. Delete the friends
    3. Search the friends' information
    4. Sort the friend list
    5. Display all the friend's information
    6. Change the friends' information
    7. Immediately store the data
    8. Close the program

3.3 Design of the sub-program



3.3.1 Read the phonebook procedure

- A text file
- File name: phone.txt
- Data stored:
  1. Name
  2. Sex
  3. Phone (8 characters)
  4. Email
  5. Remark
- File structure

Tom	-	name
M	-	sex
98765432	-	phone
tom01@gmail.com	-	email
no	-	remark

Sample file

3.3.2 Display data procedure

- Display the friend list
- Information includes name, sex, phone, email, remark
- Format:

```
Name        Sex    Phone        Email           Remark
-------------------------------------------------------------------------
XXX         X      XXXXXXXX     XXXXXXXXXX      XX
XXX         X      XXXXXXXX     XXXXXXXXXX      XX
XXX         X      XXXXXXXX     XXXXXXXXXX      XX
XXX         X      XXXXXXXX     XXXXXXXXXX      XX
XXX         X      XXXXXXXX     XXXXXXXXXX      XX
XXX         X      XXXXXXXX     XXXXXXXXXX      XX
```

3.3.3 Sort friend list procedure

- Arrange the order of friends alphabetically.
- Example:

Name	Sex	Phone	Email	Remark
CXX	X	XXXXXXXX	XXXXXXXXXX	XX
AXX	X	XXXXXXXX	XXXXXXXXXX	XX
GXX	X	XXXXXXXX	XXXXXXXXXX	XX
EXX	X	XXXXXXXX	XXXXXXXXXX	XX
ZXX	X	XXXXXXXX	XXXXXXXXXX	XX
YXX	X	XXXXXXXX	XXXXXXXXXX	XX

Name	Sex	Phone	Email	Remark
AXX	X	XXXXXXXX	XXXXXXXXXX	XX
CXX	X	XXXXXXXX	XXXXXXXXXX	XX
EXX	X	XXXXXXXX	XXXXXXXXXX	XX
GXX	X	XXXXXXXX	XXXXXXXXXX	XX
YXX	X	XXXXXXXX	XXXXXXXXXX	XX
ZXX	X	XXXXXXXX	XXXXXXXXXX	XX

Chapter 4: Implementation

4.1 Brief Description

In this chapter, I will discuss the implementation of the phonebook directory program.

I will:

1. To develop the program by using Dev-Pascal
2. To compile the source program into the object program
3. Describe the functions that will be performed by each procedure in the program.

4.2 Procedures in the program

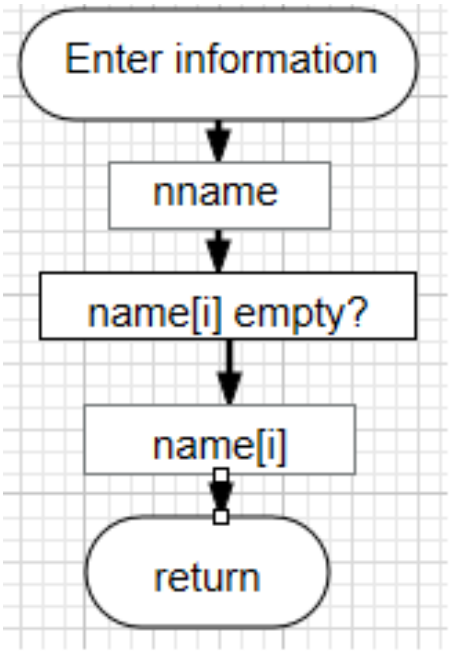
I will use the following parallel arrays to store the friends' name, sex, phone, email and some remark of them:

```
name : array[1..max] of string;
sex : array[1..max] of string;
phone : array[1..max] of string[8];
email : array[1..max] of string;
remark : array[1..max] of string;
```

where max is a constant defined at the beginning of the program

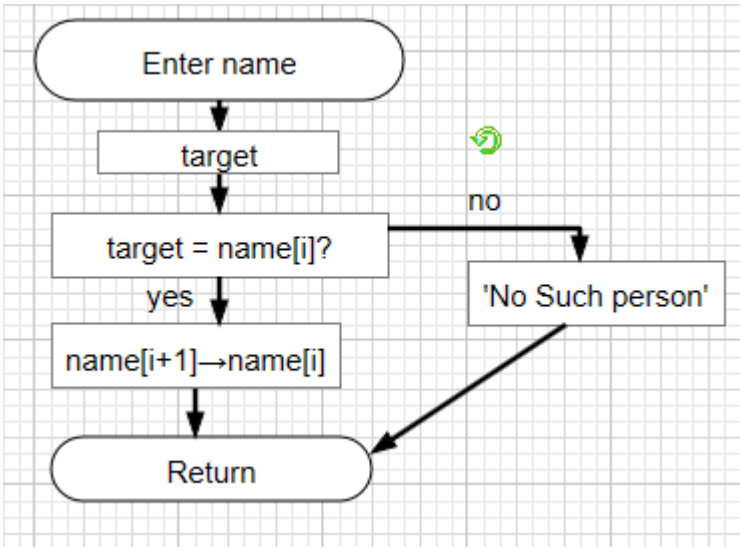
```
const max = 200;
```

4.2.1 Add new friends



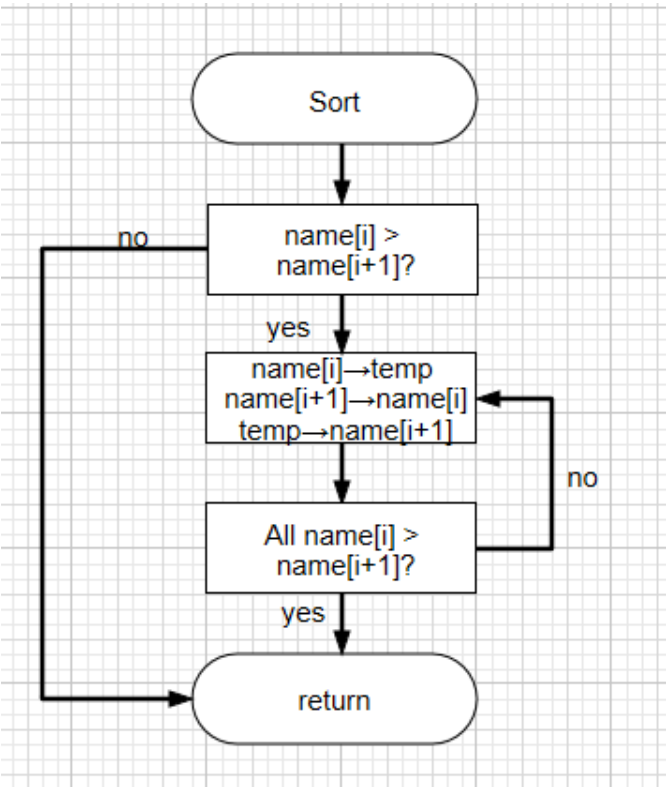
In this procedure, ‘nname’ is a temporary variable to store the entered data, for entered sex, phone, email and remark will store in ‘nsex’, ‘nphone’, ‘nemail’ and ‘nremark’ respectively.
Then it will find an empty array, and the empty array will store the new entered data.

4.2.2 Delete the friends



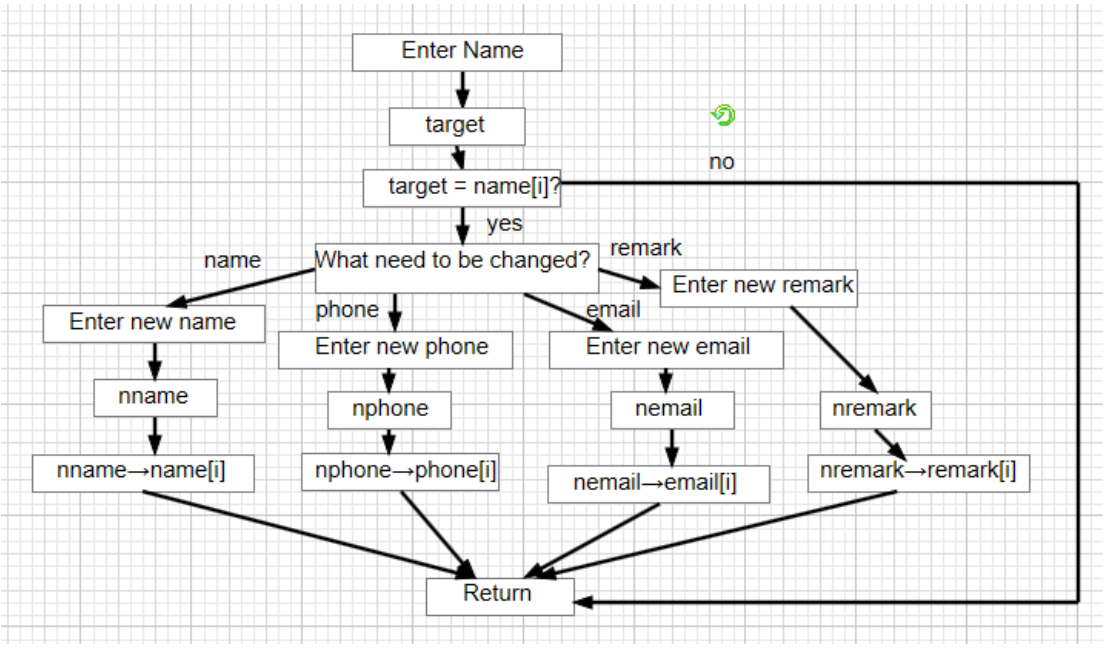
In the procedure, user will first enter a name which he/ she want to delete that information, the entered name will store in a variable called ‘target’, if the target meet the friend’s name, and the data of this friend will all be deleted, otherwise the words ‘No such person’ will appear.

4.2.3 Sort the data



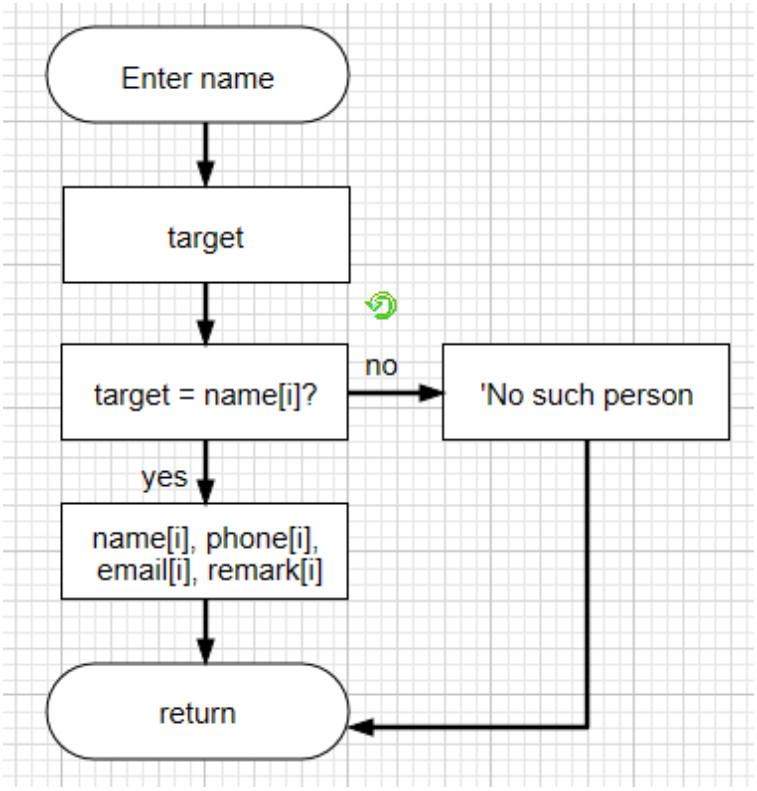
The program will arrange the data alphabetically according to the name, and the way of sorting in this procedure is bubble sort. First some temporary variables will store the first array, and compare with the first letter of the name. For example, there are two names in the list, which are ‘Tom’ and ‘Jason’, in the ASCII, the code of ‘J’ is bigger than ‘T’, so in the list, the order of ‘Tom’ and ‘Jason’ will swap. The sorting will be until all the friends’ name is in alphabetical order.

4.2.4 Change the data



In this procedure, user will first enter a name which he/ she want to change that information, the entered name will store in a variable called ‘target’, if the target meet the friend’s name, and his / her information will be listed out. And the users can select what he or she wants to change. The entered name, sex, phone, email and remark will store in temporary variables ‘nname’, ‘nsex’, ‘nphone’, ‘nemail’ and ‘nremark’ respectively. Then the recent array will be changed to the new entered data.

4.2.5 Search friends

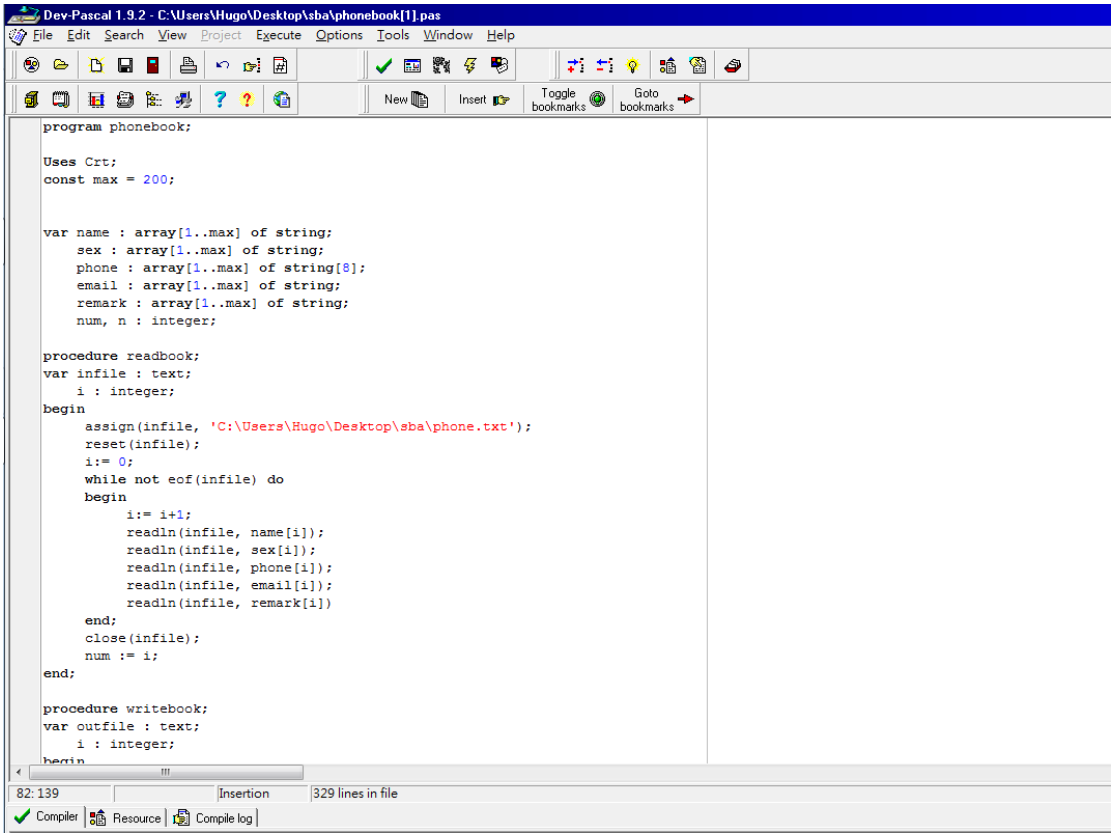


In this procedure, user will first enter a name which he/ she want to change that information, the entered name will store in a variable called ‘target’, if the target meet the friend’s name, and his / her information will be listed out.

4.3 Program coding

In this project, I will use Pascal program called ‘Dev-Pascal’.

The file name of the phonebook is ‘phone.txt’.

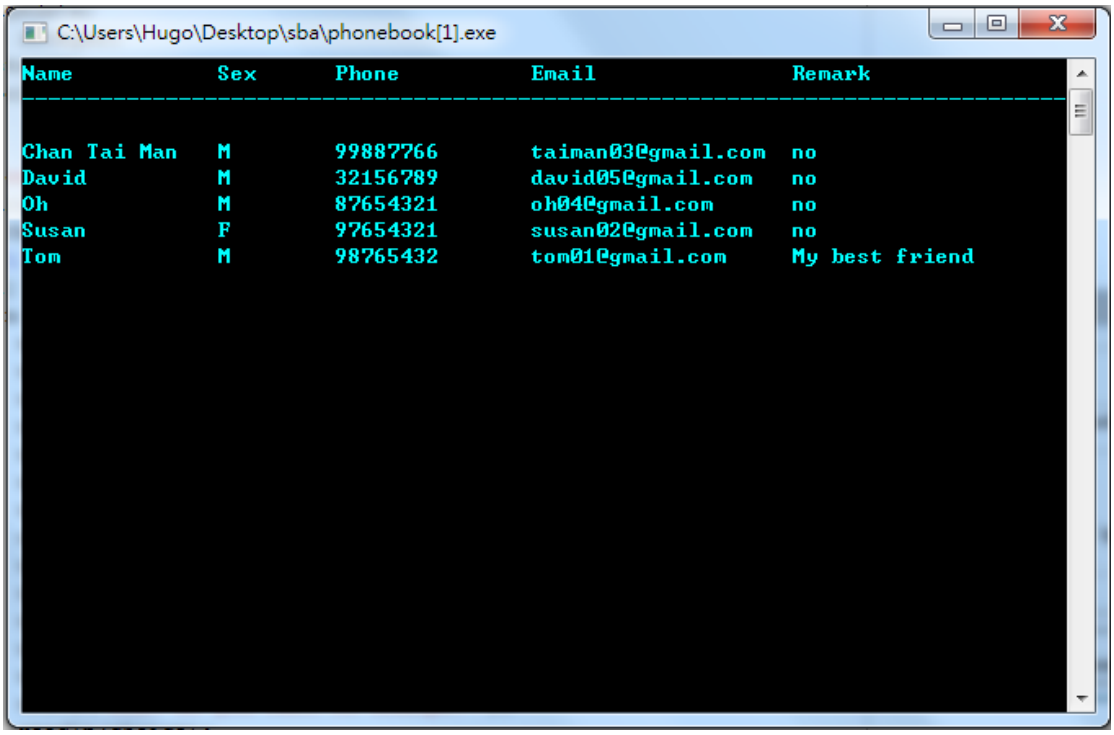


The complete program code can refer to Appendix 1.

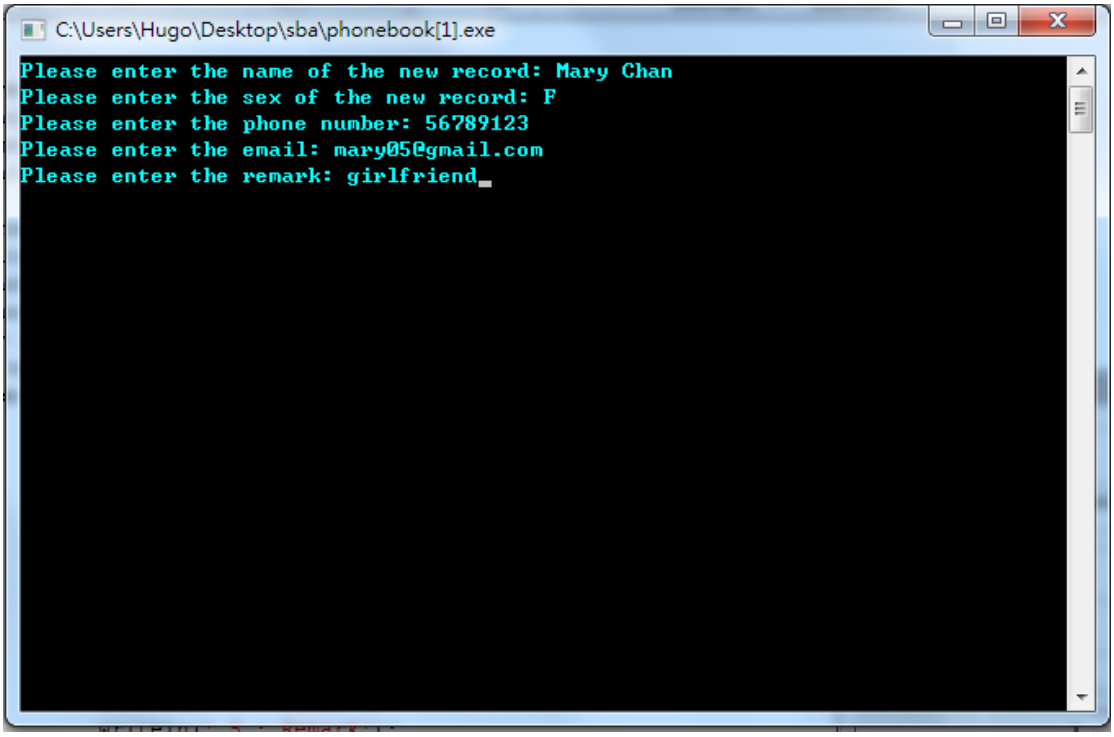
4.4 Program execution

In the program, the program name is phonebook.exe.
The phonebook file type is '.txt' and the file name is called 'phone.txt'.

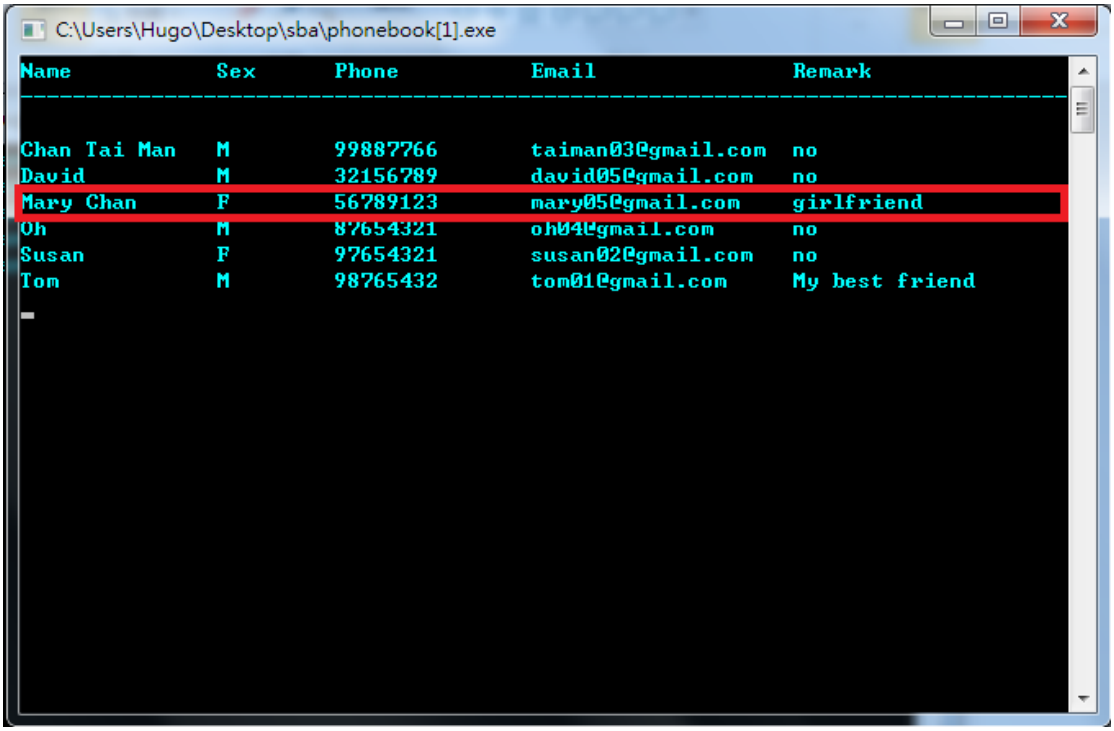
Test file



4.4.1 Add the new friends

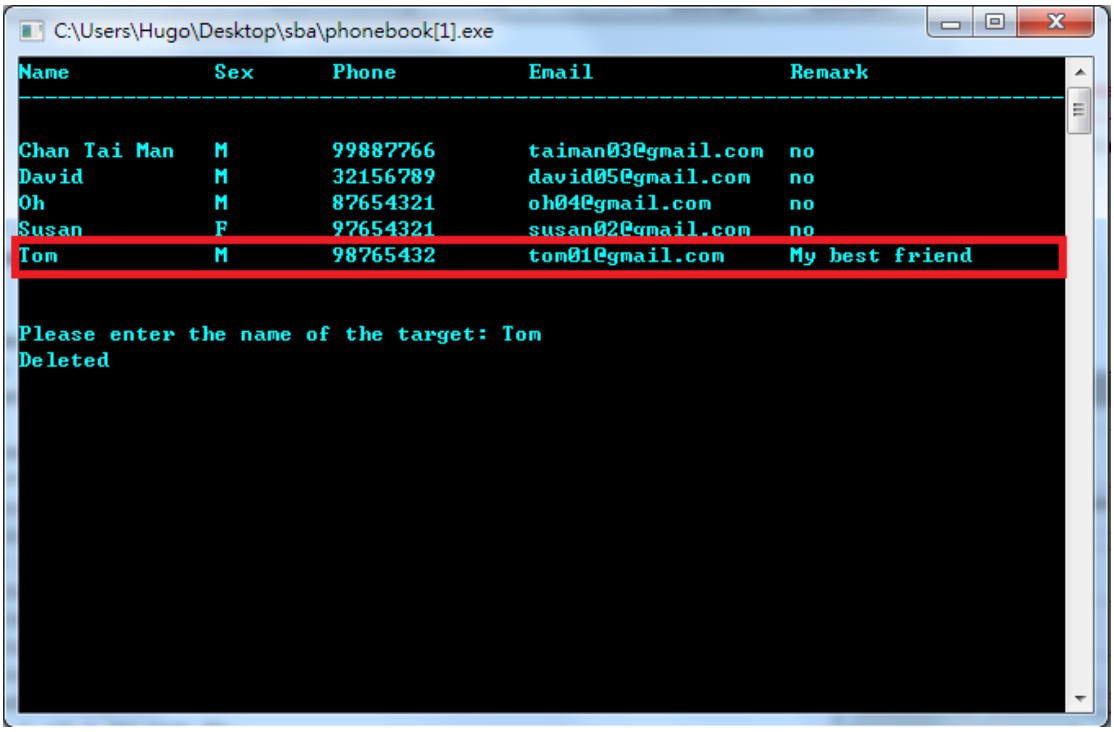


After

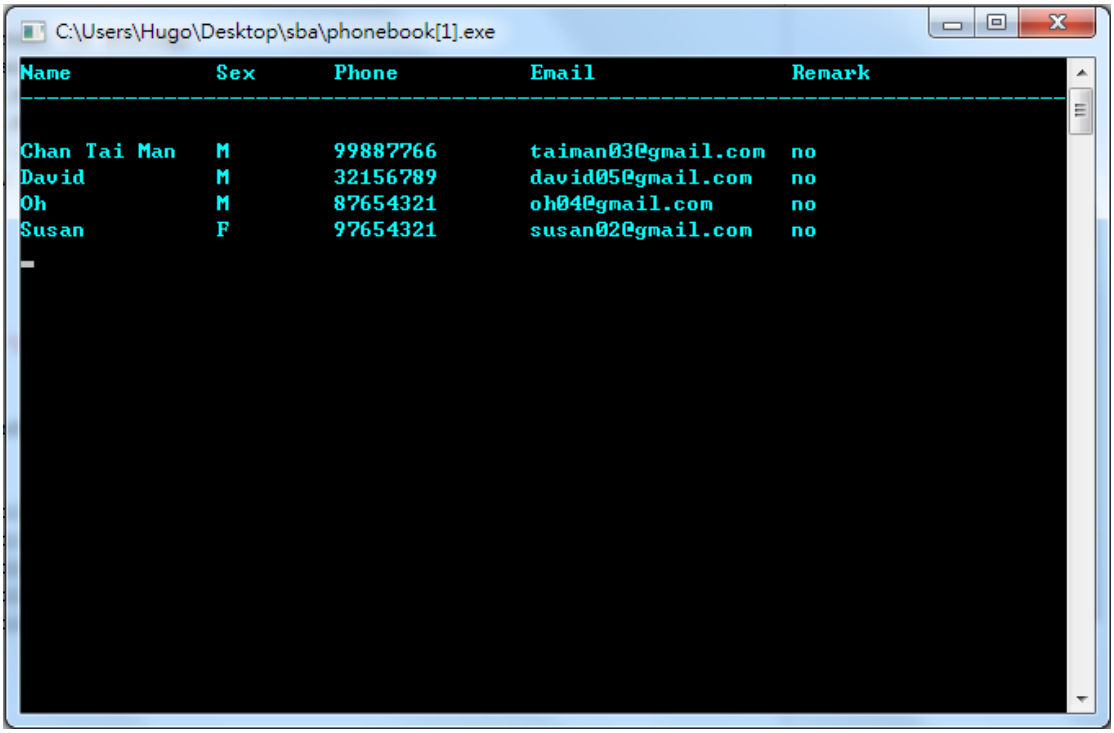


A new friend (Mary Chan) is added.

4.4.2 Delete the friends



After

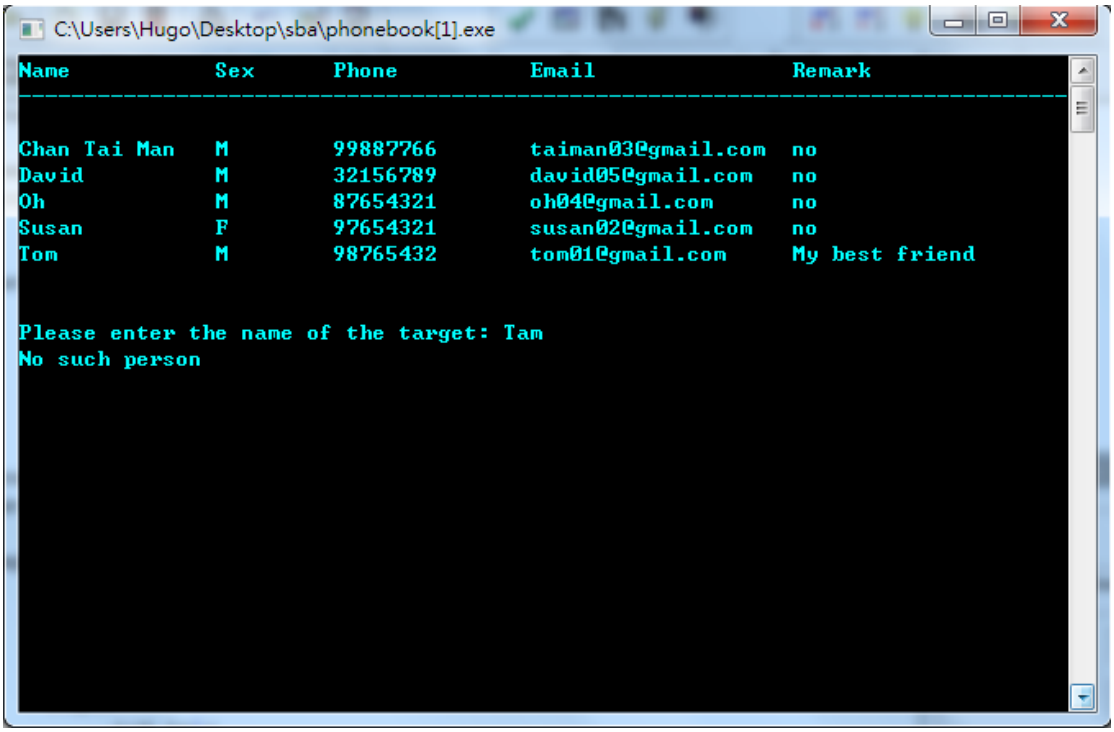


Friend (Tom) is deleted.

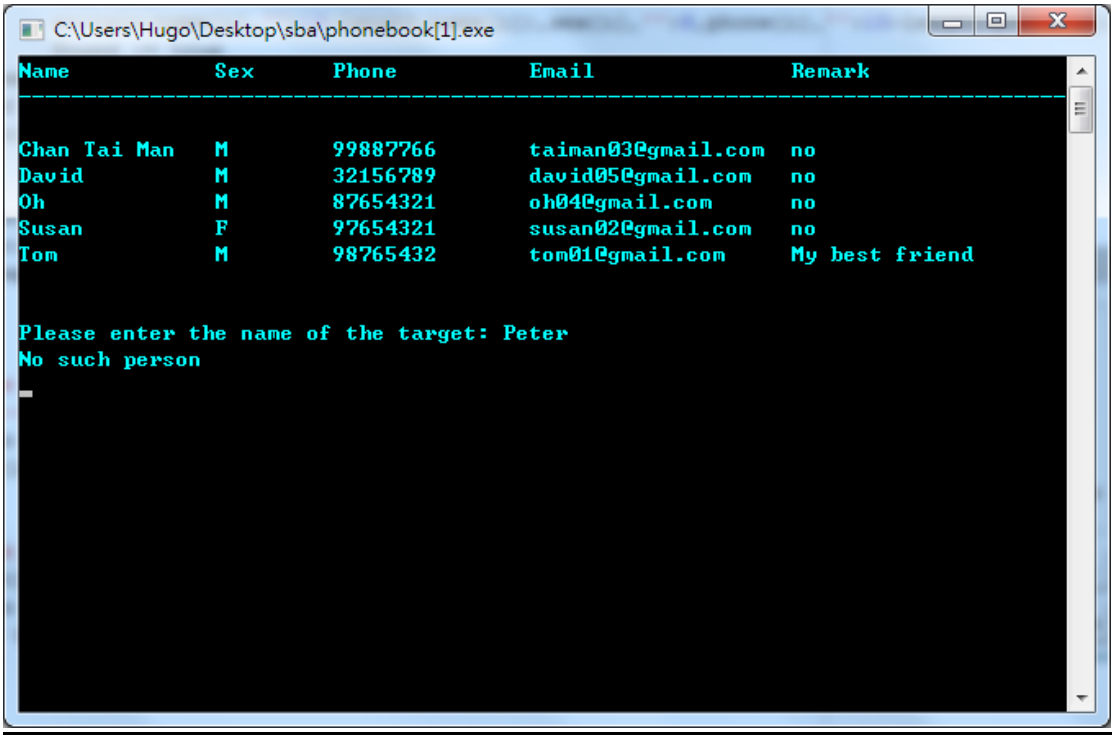
But if the target is wrongly typed or there isn't the friend, the words 'No such person'

will appear.

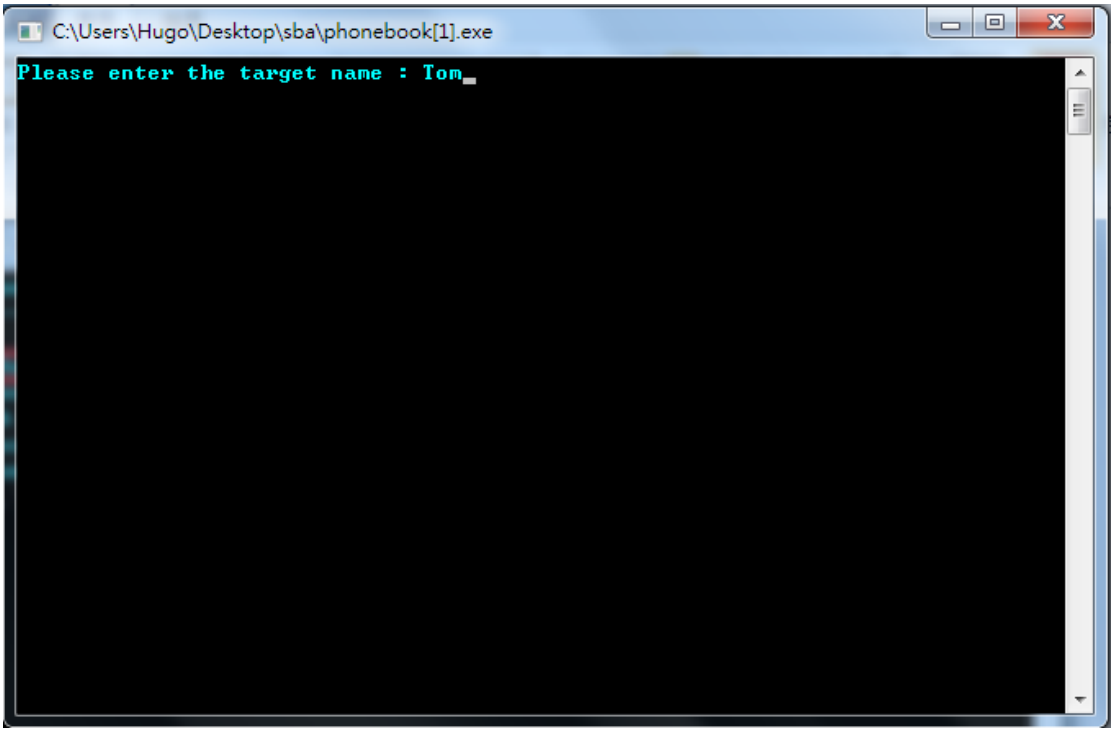
Target wrongly typed



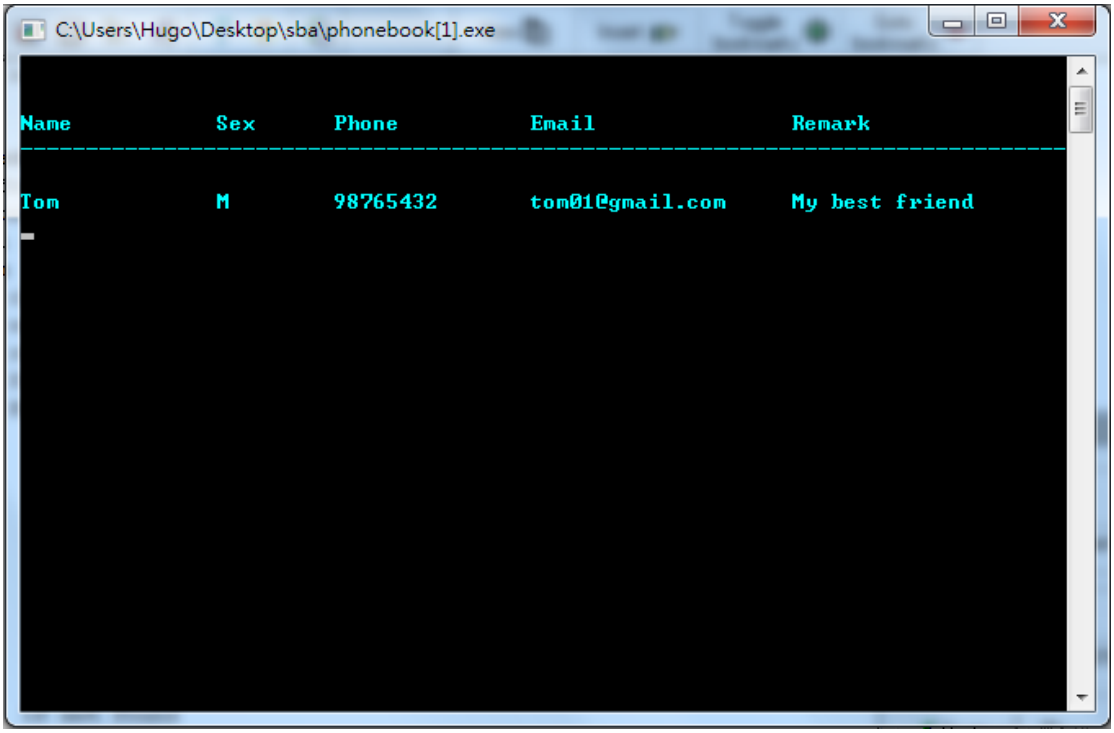
Target not existing



4.4.3 Search the friends

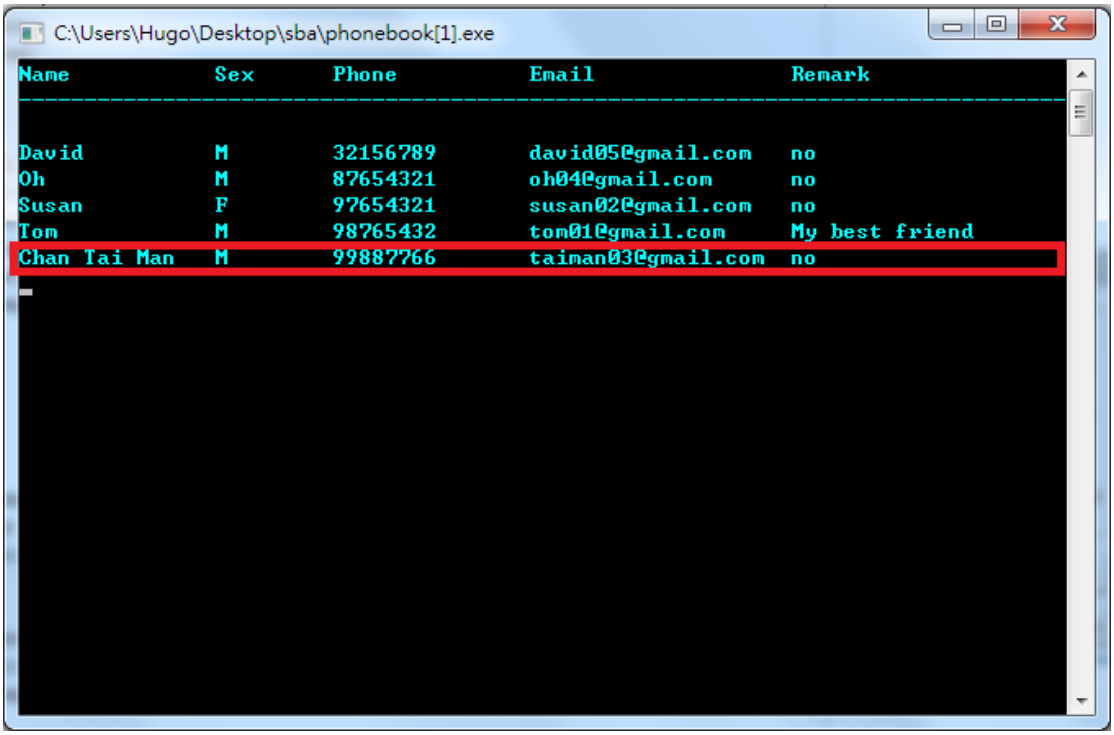


After

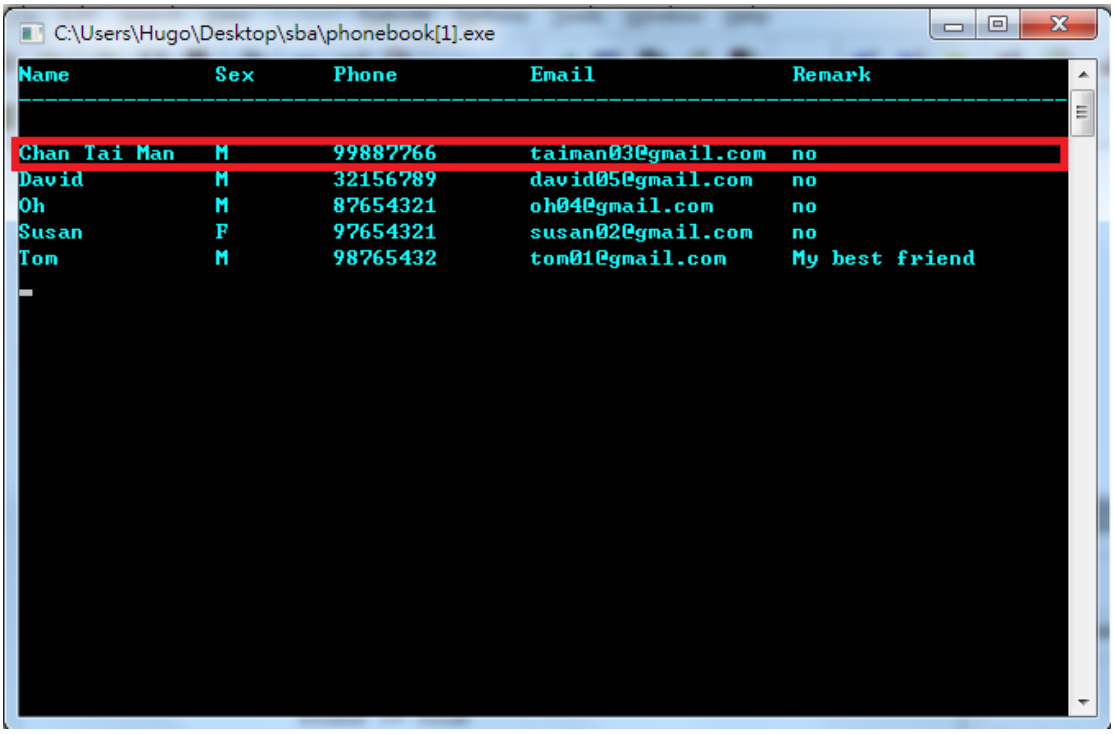


4.4.4 Sort the friend list

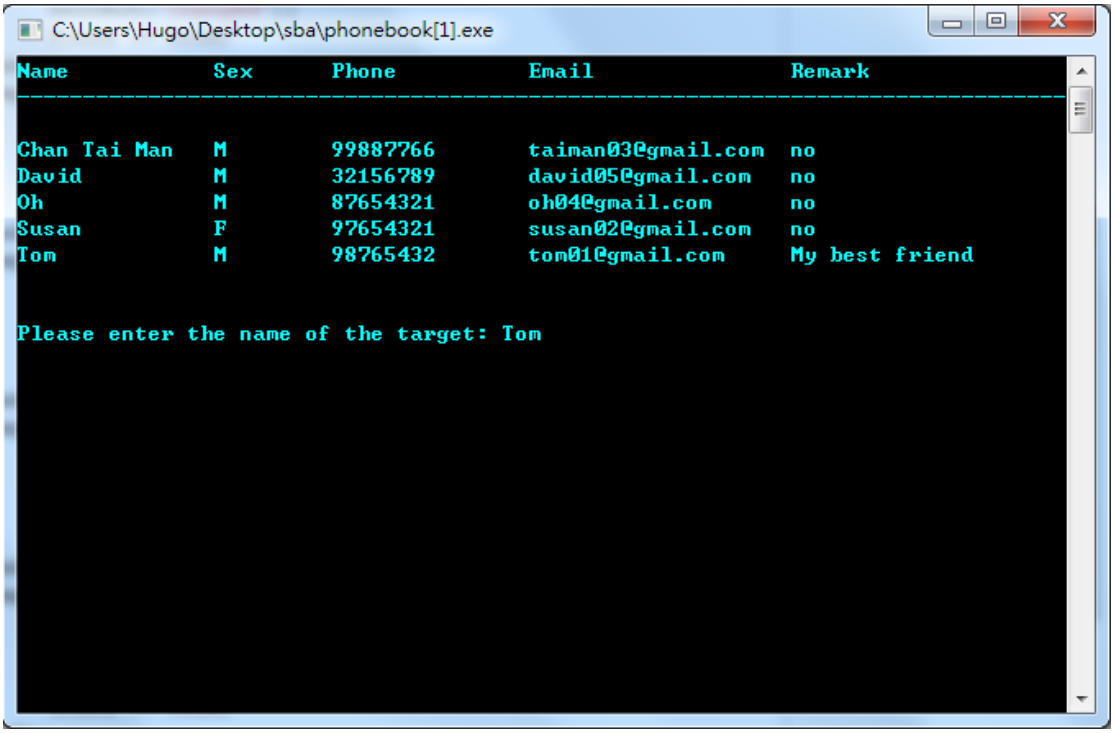
Before



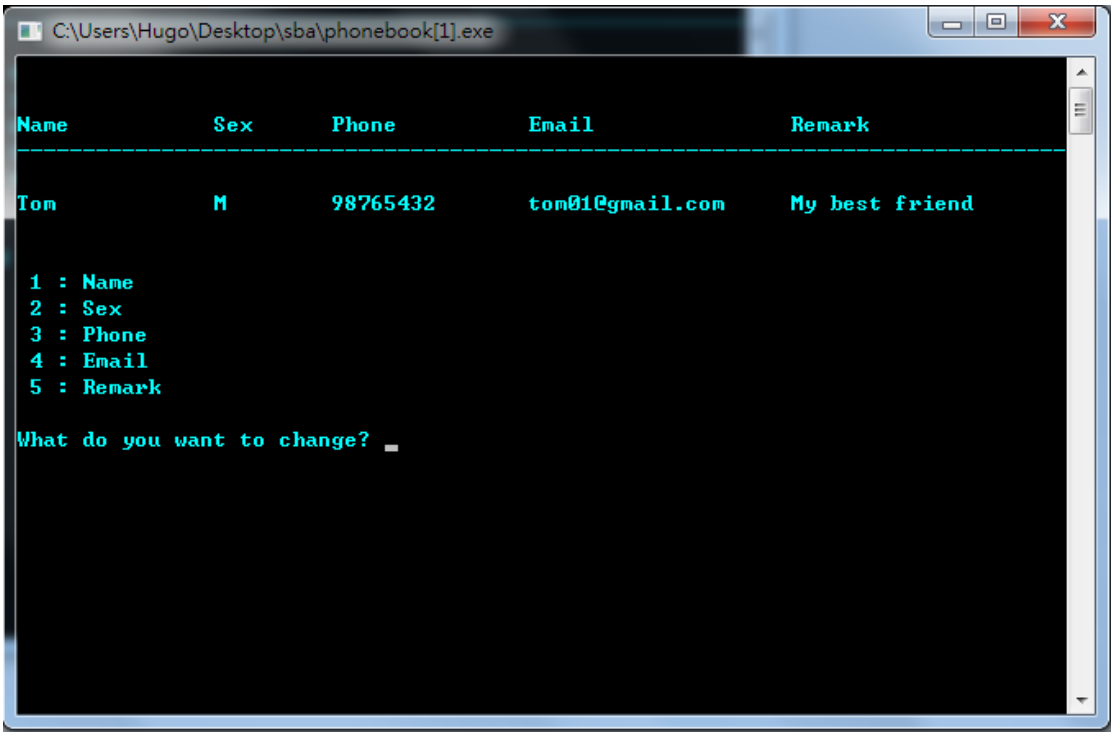
After



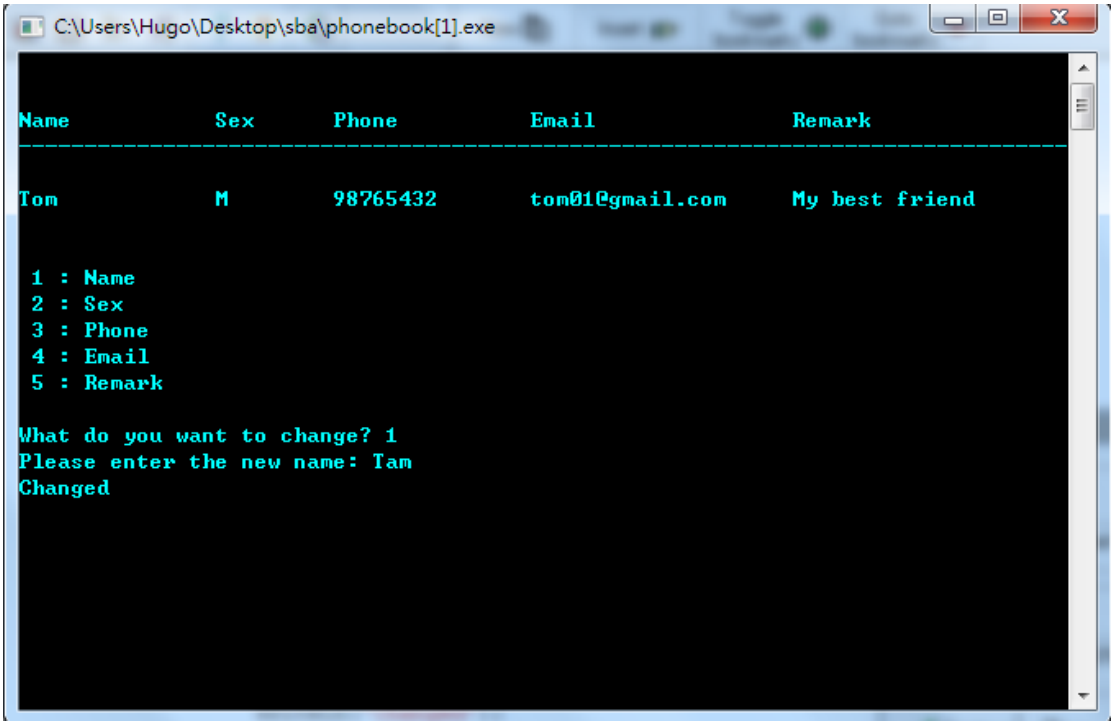
4.4.5 Change the friends' information



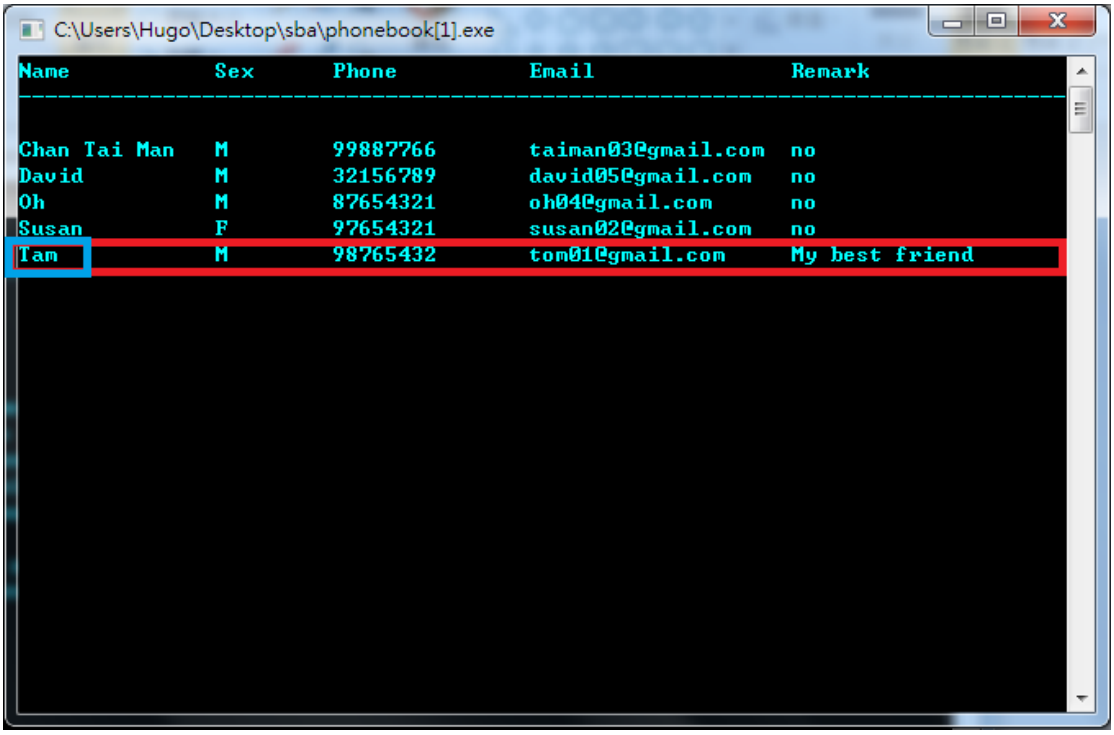
After the target found



Change the name

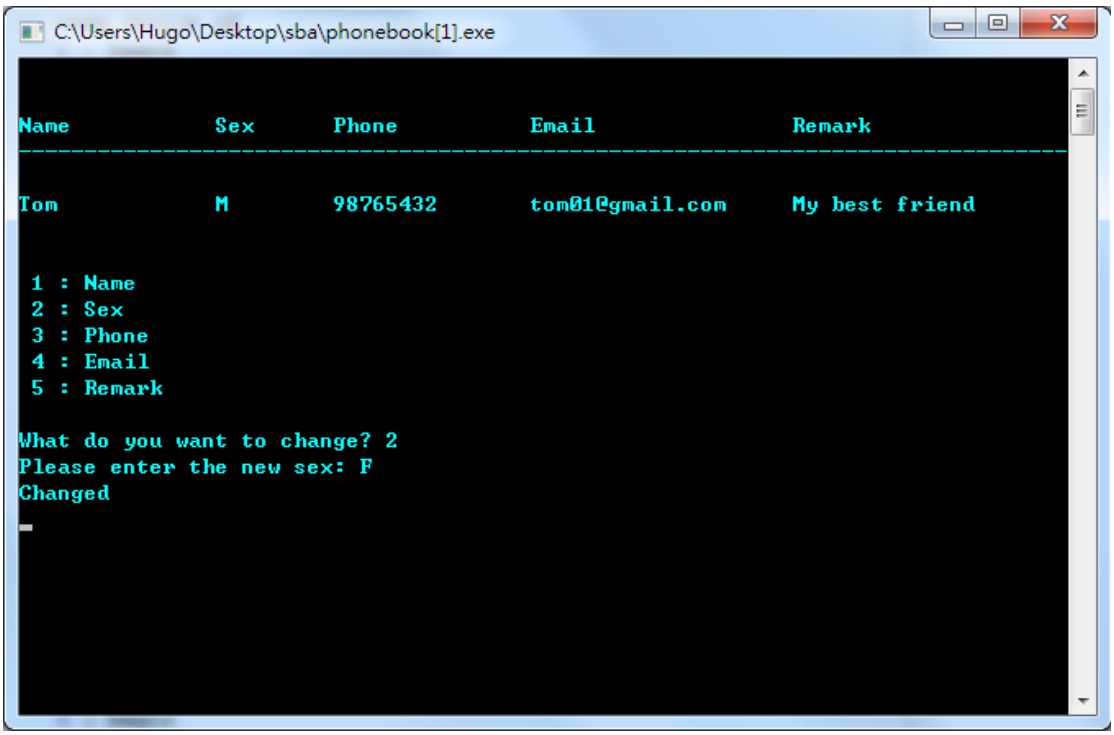


After

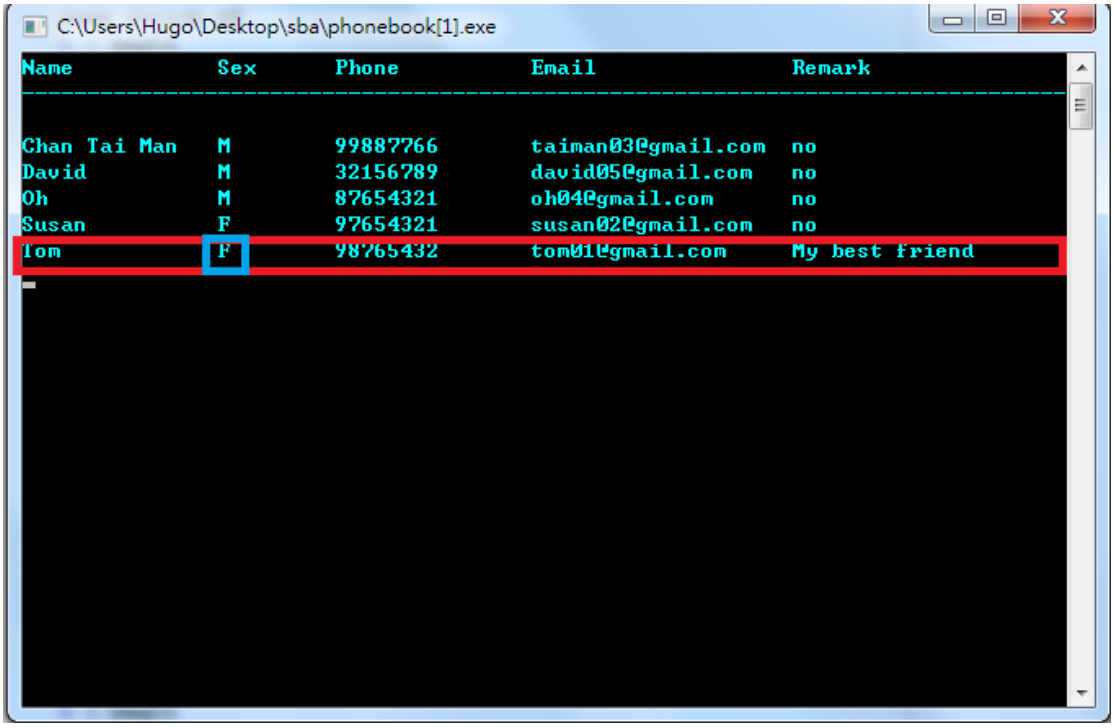


The name is changed.

Change the sex

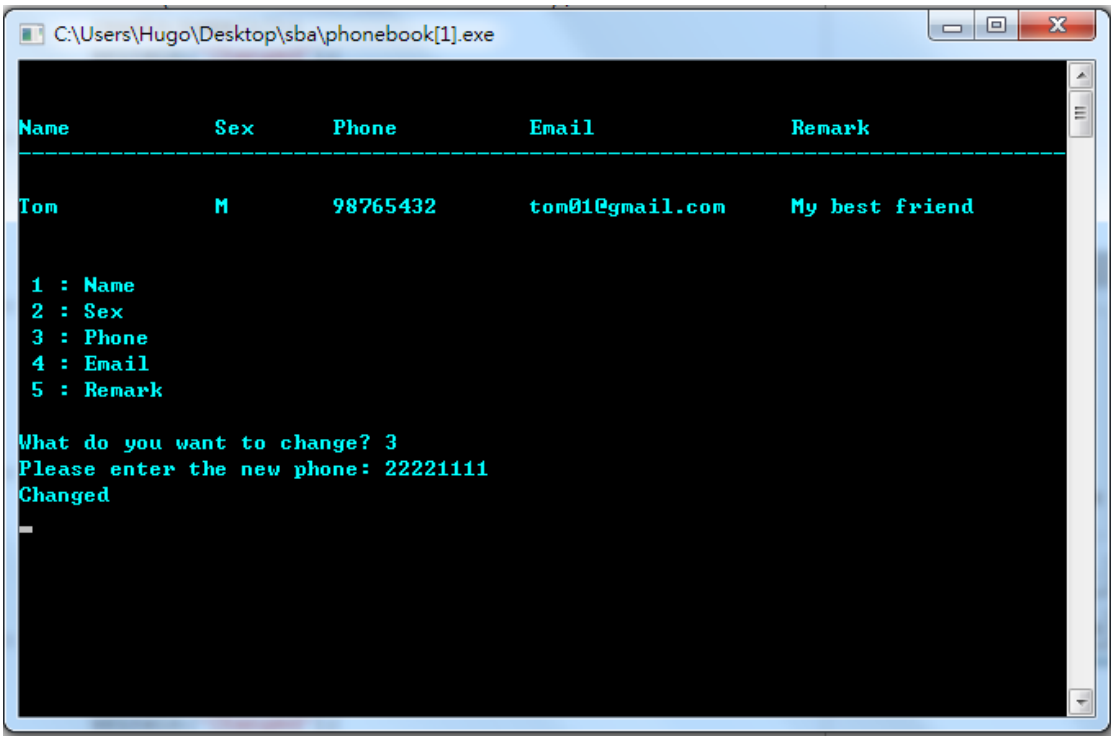


After

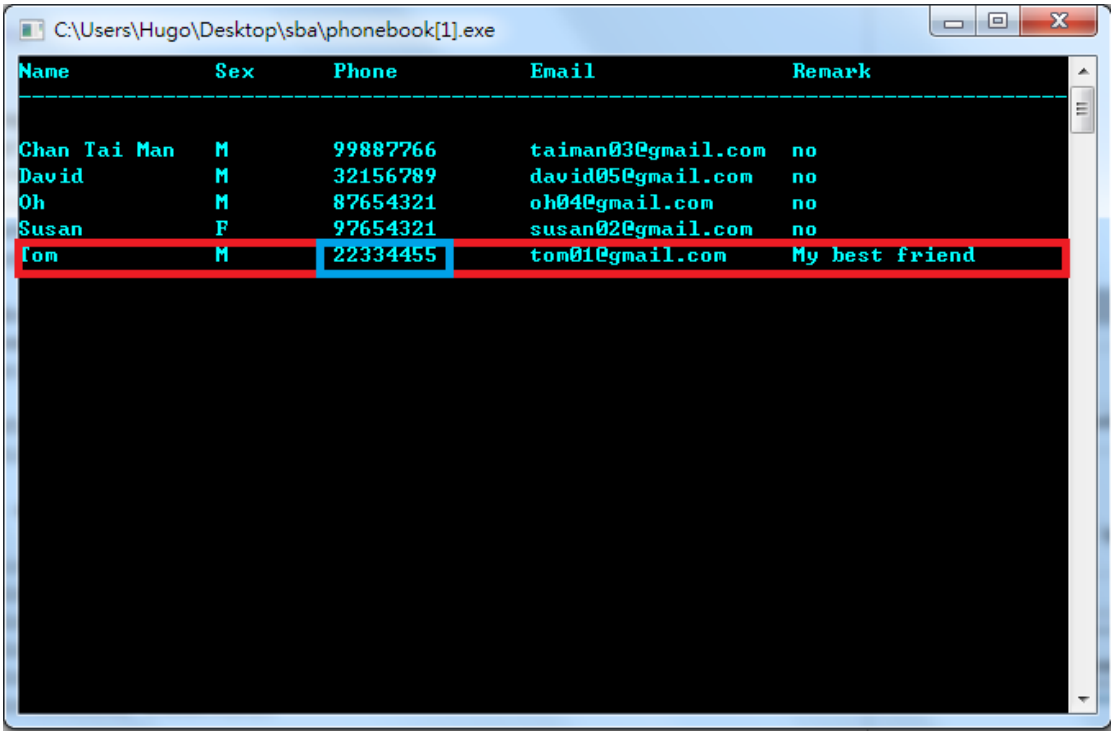


The sex is changed.

Change the phone number

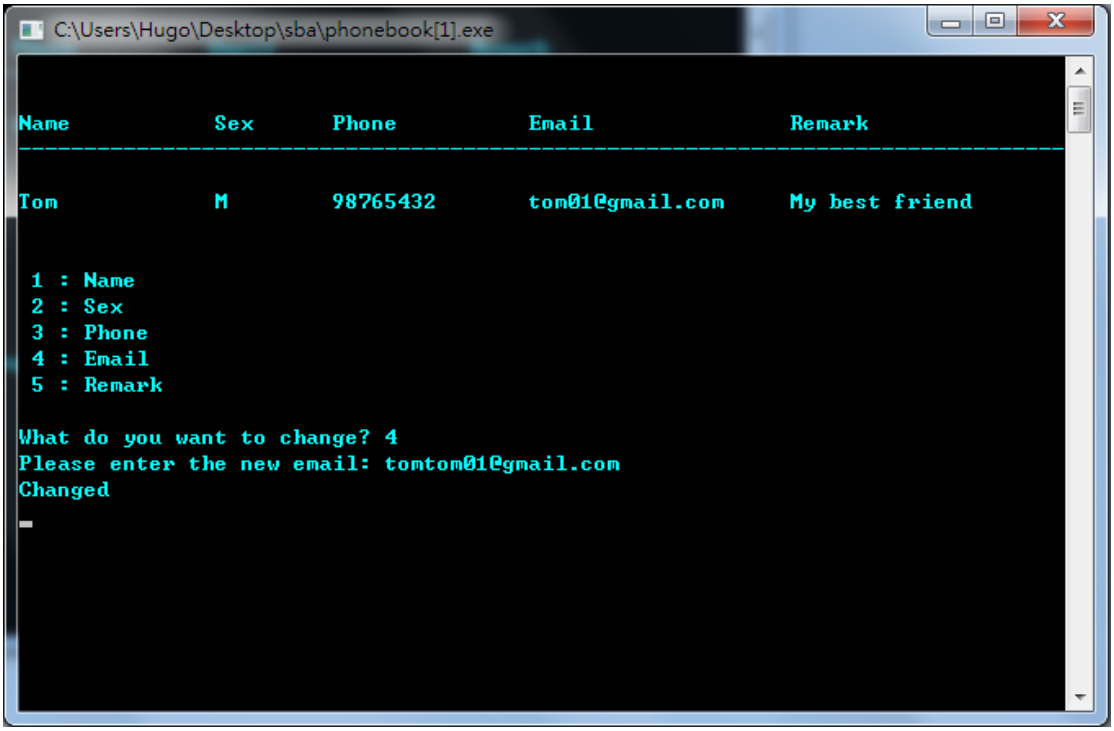


After

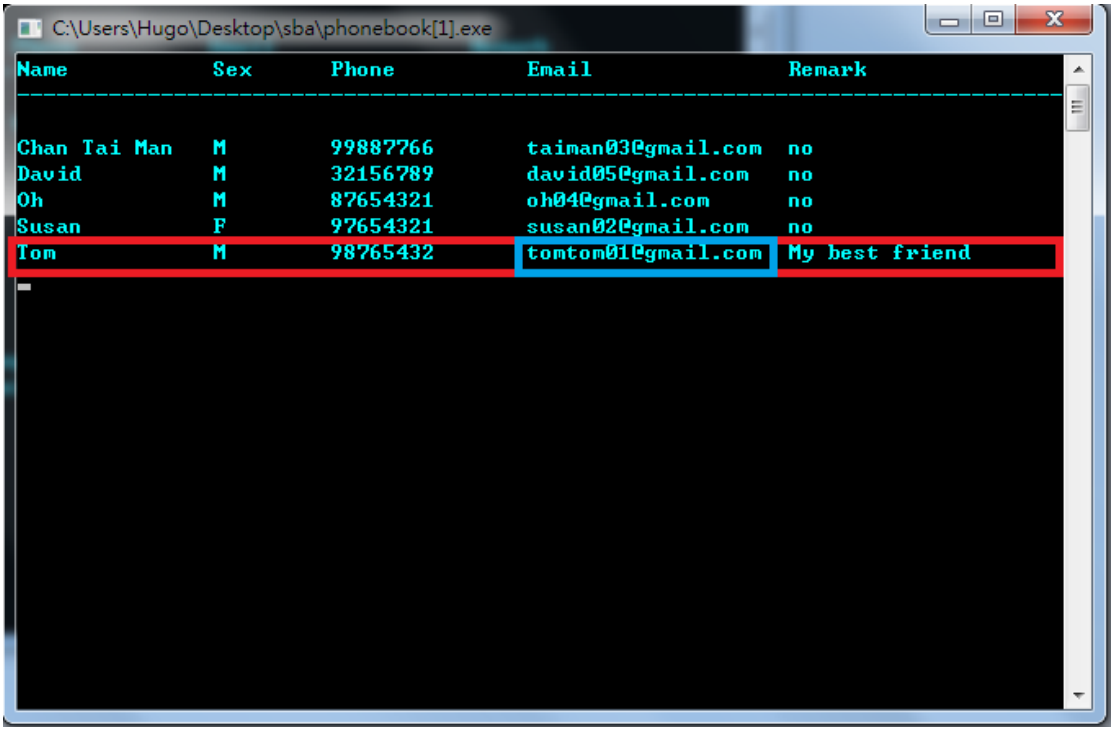


The phone number is changed.

Change the email

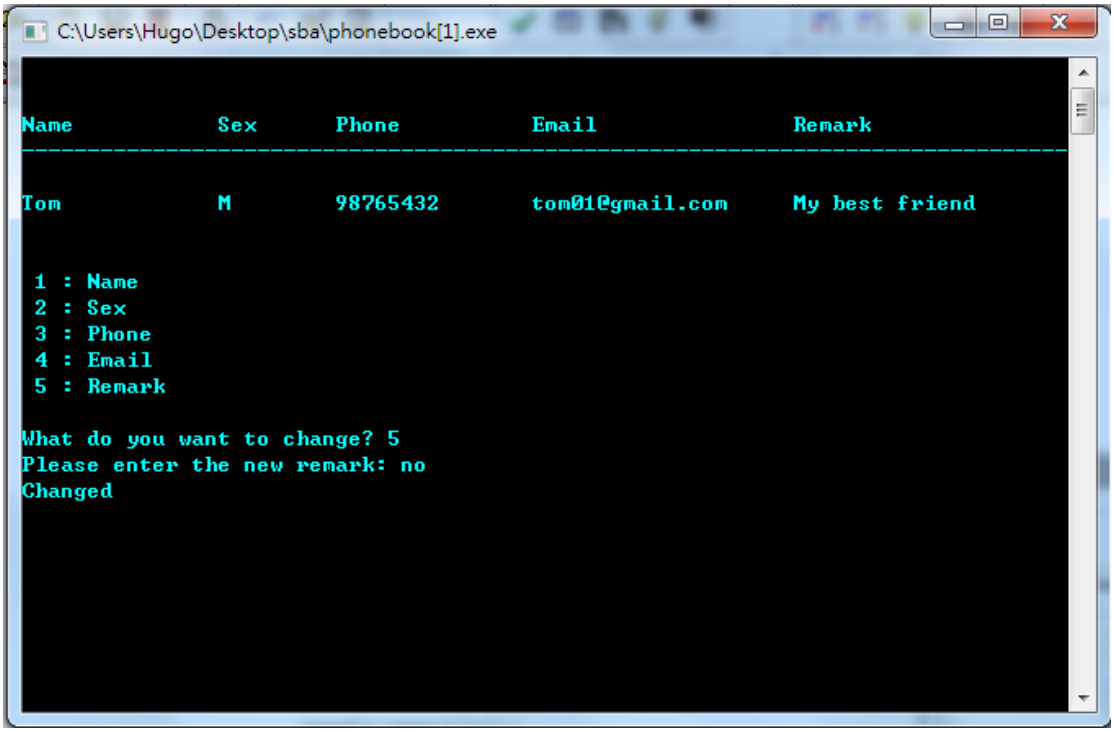


After

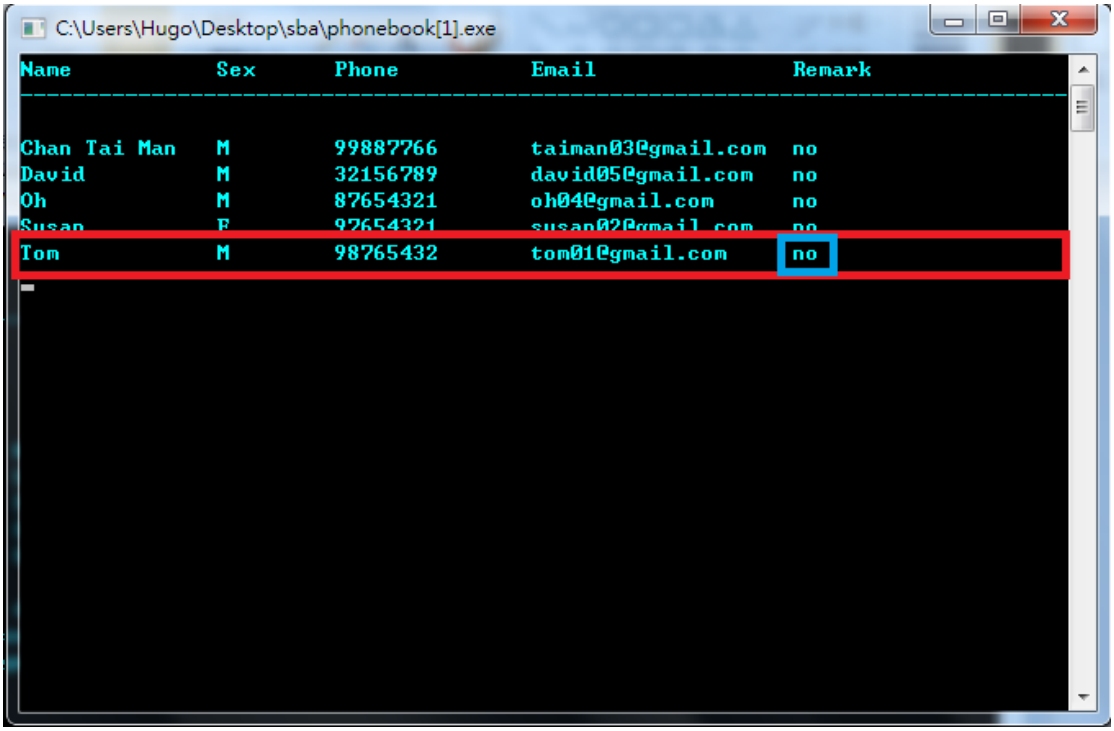


The email is changed.

Change the remark



After



The remark is changed.

Chapter 5: Testing & Evaluation

5.1 Brief Description

- To find out the bugs (logical and run-time errors) in the program
- To check whether the program can achieve its purposes
- To debug and improve the program based on the testing and evaluation results

5.2 Testing and Evaluation Plan

The program will be tested and evaluated according to the following plan:

1. Internal testing and evaluation / Tested and evaluated by me (the programmer):

- The program will be tested intensively by me – the programmer
- I will prepare some test cases to test the program thoroughly
- The test cases include some correct input data (from files & keyboard) with known results for checking the correctness of the program, some incorrect input data to see whether the program can handle invalid input reasonably.

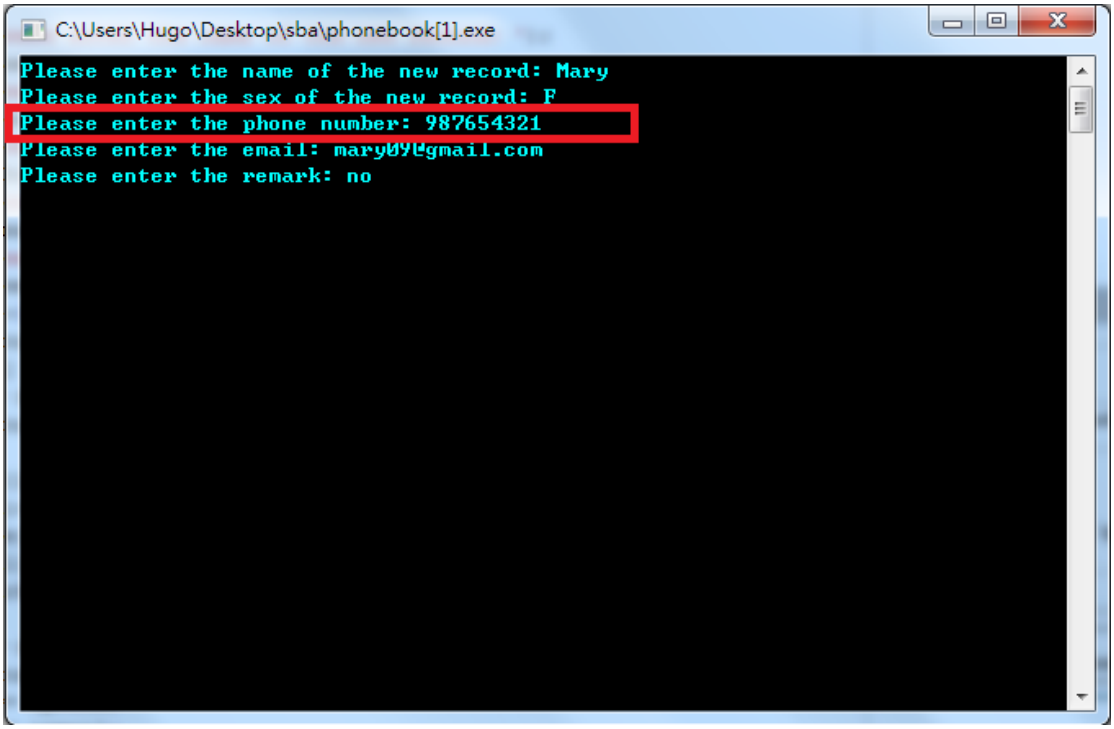
5.3 Internal Testing

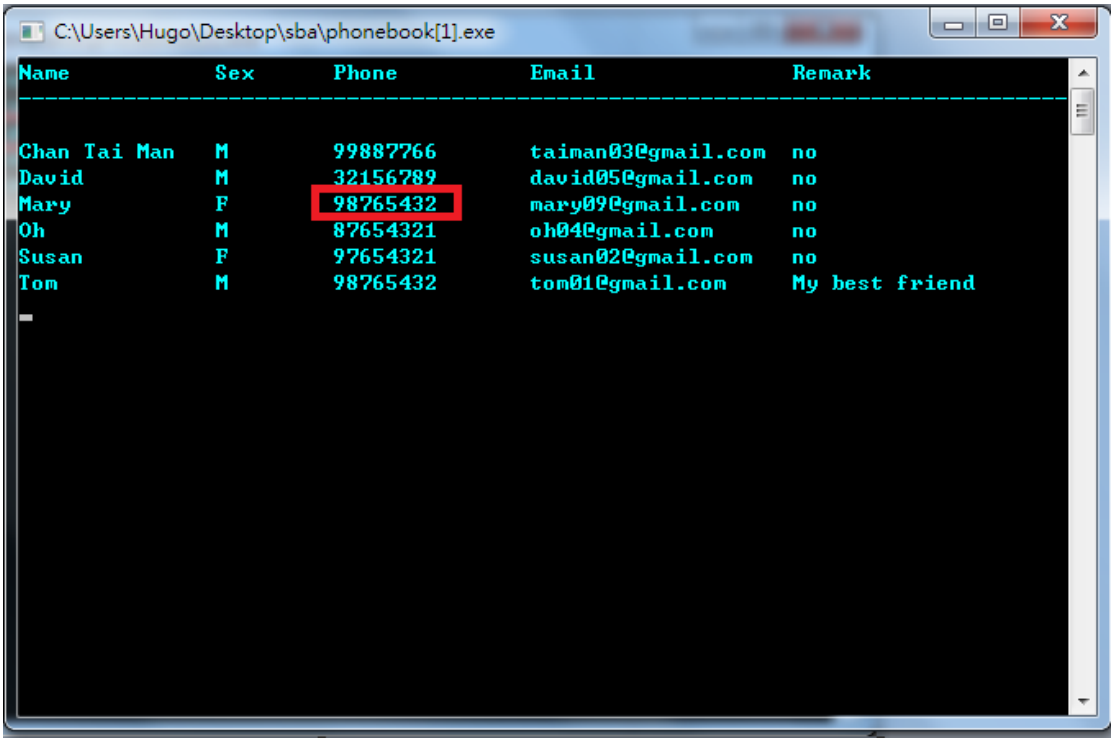
Test case 1

Purpose:	To check whether the phone number can be limited as 8 characters
Input:	9 and 10 characters of phone number
Expected Output:	The data will not be stored
Actual Output:	First 8 characters of the phone number has input, the remainder disappear
Test Result:	Bug found
Follow-up Action:	Develop a repeat loop for asking user to rewrite phone number if there are more than 8 characters of the entered phone number

Before

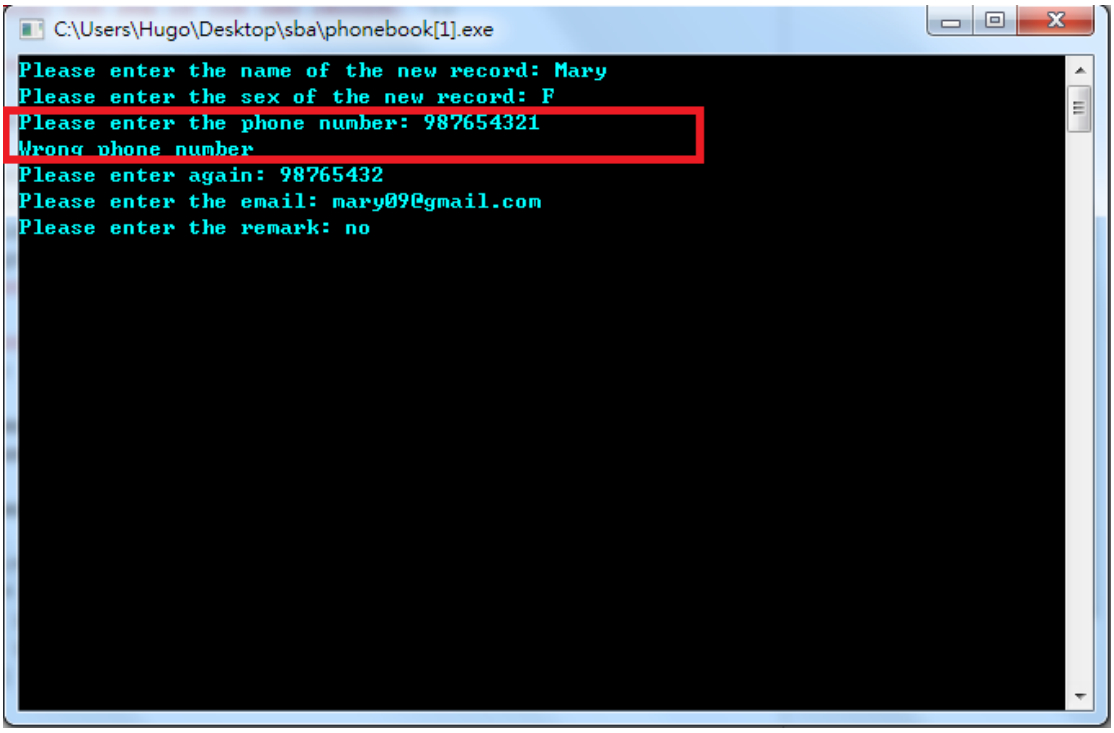
```
begin
     clrscr;
     write('Please enter the name of the new record: ');
     readln(nname);
     write('Please enter the sex of the new record: ');
     readln(nsex);
     write('Please enter the phone number: ');
     readln(nphone);
     write('Please enter the email: ');
     readln(nemail);
     write('Please enter the remark: ');
     readln(nremark);
     n := num +1;
     for i := num downto 1 do
     if name[i] > nname
     then n := n-1;
     for i := num downto n do
     begin
          name[i+1] := name[i];
          sex[i+1] := sex[i];
          phone[i+1] := phone[i];
          email[i+1] := email[i];
          remark[i+1] := remark[i];
     end;
     name[n]:= nname;
     sex[n]:= nsex;
     phone[n]:= nphone;
     email[n]:= nemail;
     remark[n]:= nremark;
     num := num+1;
end;
```





After the follow-up action

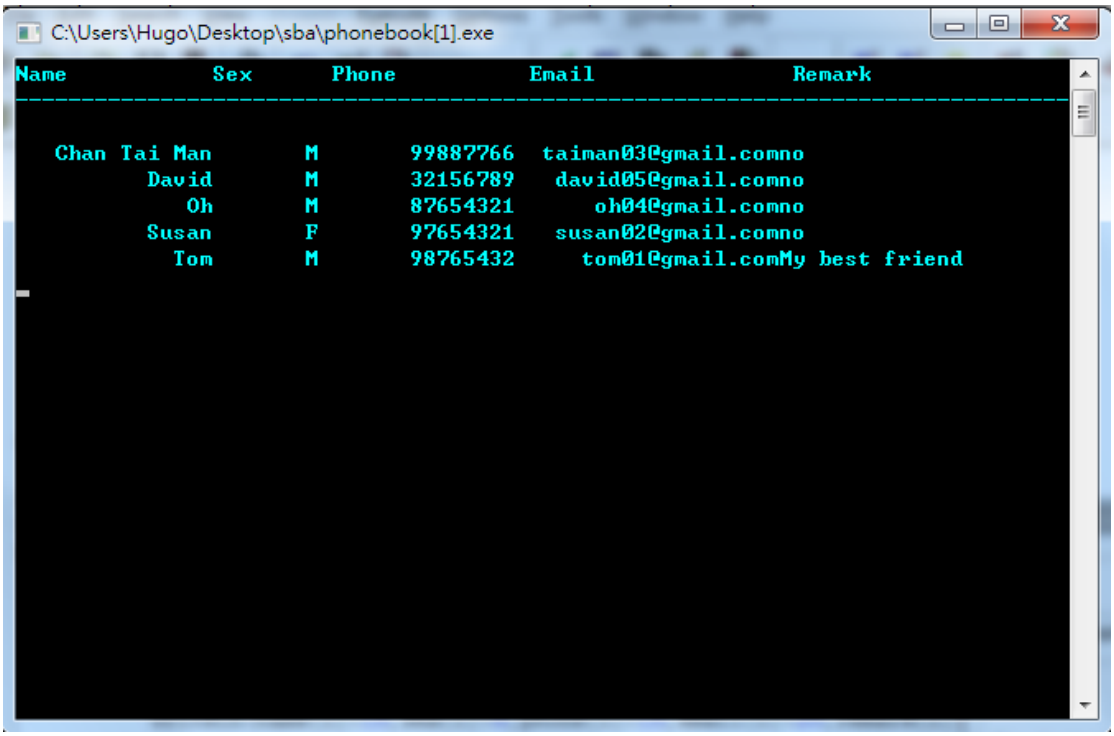
```
begin
      clrscr;
      write('Please enter the name of the new record: ');
      readln(nname);
      write('Please enter the sex of the new record: ');
      readln(nsex);
      write('Please enter the phone number: ');
      readln(nphone);
      if length(nphone)>8
      then repeat
                  writeln('Wrong phone number');
                  write('Please enter again: ');
                  readln(nphone);
           until (length(nphone)<=8);
      write('Please enter the email: ');
      readln(nemail);
      write('Please enter the remark: ');
      readln(nremark);
      n := num +1;
      for i := num downto 1 do
      if name[i] > nname
      then n := n-1;
      for i := num downto n do
```



Test case 2

Purpose:	To test whether data can display tidily
Input:	Normal data
Expected Output:	Data can tidily display
Actual Output:	Untidy displayed data
Test Result:	Need to improve
Follow-up Action:	Control the value of the space between information

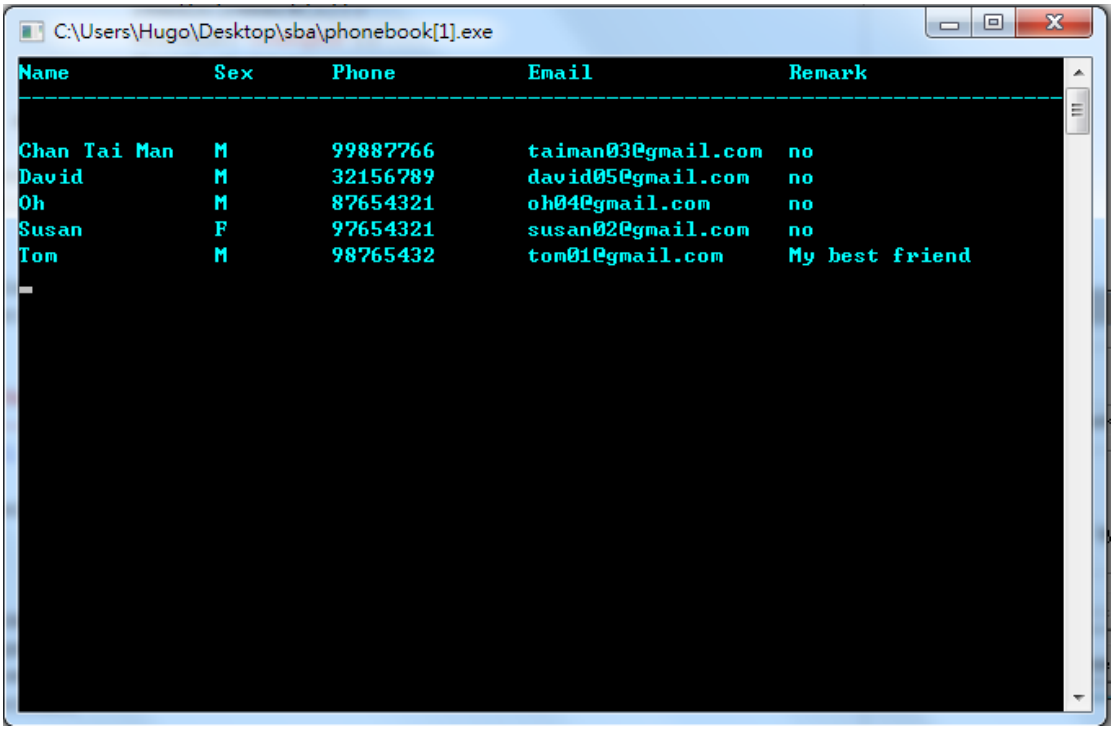
```
procedure displaydata;
var i : integer;
begin
     clrscr;
     writeln('Name           Sex      Phone          Email               Remark' );
     writeln('-------------------------------------------------------------------------------' );
     for i := 1 to num do
     begin
          writeln(name[i]:15,sex[i]:8,phone[i]:15,email[i]:20,remark[i])
     end
end;
```



After follow-up action

```
procedure displaydata;
var i : integer;
begin
     clrscr;
     writeln('Name          Sex      Phone         Email             Remark' );
     writeln('-------------------------------------------------------------------------------' );
     for i := 1 to num do
     begin
          writeln(name[i],'':15-length(name[i]),sex[i],'':8,phone[i],'':15-length(phone[i]),email[i],'':20-length(email[i]),remark[i])
     end
end;
```

‘writeln(name[i],'':15-length(name[i]),sex[i],'':8,phone[i],'':15-length(phone[i]),email[i],'':20-length(email[i]),remark[i])’



5.4 Self-Evaluation

- User interface is simple but users can easily control and use it.
- For phone number, it can reduce the opportunity of filling wrong phone number (Characters of the phone number)
- Number of function is a bit less, but it contains the basic functions
- The flexibility of future development is high, for example, the log-in function can add in the program.

Chapter 6: Conclusion & Discussion

In this course work, my topic is phone book directory program, normal phone book contain name and phone, for my designed phone book directory program, it contains name, sex, phone, email and remark, users can add more information about their friends.

In the program, users can add and delete friends, also, users can search their friends' information, display the information and change it. For tidying up the friend list, users can also use the sorting function.

Although those functions are basic, it helps users simply manage their friend list, and I have designed the user interface as simply as possible, all users can learn to use it easily.

Through this course work, it helps me recall the memory of many programming techniques and I try to develop a program, from analysis to testing, also, while searching information about the course work, I learn some new techniques, such as correct some logical and run-time error.

Chapter 7: Reference and Acknowledgment

7.1 Reference

- http://www.psdgraphics.com/file/phone-book-icon.jpg
- http://ts4.mm.bing.net/th?id=H.4965357087688343&pid=15.1
- http://www.i-garden.org/archive/v2/docu/cecs/pas11.pdf

7.2 Acknowledgment

Special thanks to my ICT teacher, Mr. Chu and my classmates.

Appendix 1: Program Code

```
program phonebook;

Uses Crt;
const max = 200;


var name : array[1..max] of string;
      sex : array[1..max] of string;
      phone : array[1..max] of string[8];
      email : array[1..max] of string;
      remark : array[1..max] of string;
      num, n : integer;

procedure readbook;
var infile : text;
      i : integer;
begin
       assign(infile, 'C:\Users\Hugo\Desktop\sba\phone.txt');
       reset(infile);
       i:= 0;
       while not eof(infile) do
       begin
              i:= i+1;
              readln(infile, name[i]);
              readln(infile, sex[i]);
              readln(infile, phone[i]);
              readln(infile, email[i]);
              readln(infile, remark[i])
       end;
       close(infile);
       num := i;
end;

procedure writebook;
var outfile : text;
      i : integer;
begin
       assign(outfile, 'C:\Users\Hugo\Desktop\sba\phone.txt');
```

```
        rewrite(outfile);
        for i := 1 to num do
        begin
                writeln(outfile, name[i]);
                writeln(outfile, sex[i]);
                writeln(outfile, phone[i]);
                writeln(outfile, email[i]);
                writeln(outfile, remark[i]);
        end;
        close(outfile)
end;

procedure adddata;
var nname, nsex, nphone, nemail, nremark : string;
      i,n : integer;
begin
        clrscr;
        write('Please enter the name of the new record: ');
        readln(nname);
        write('Please enter the sex of the new record: ');
        readln(nsex);
        write('Please enter the phone number: ');
        readln(nphone);
        if length(nphone)>8
        then repeat
                        writeln('Wrong phone number');
                        write('Please enter again: ');
                        readln(nphone);
                until (length(nphone)<=8);
        write('Please enter the email: ');
        readln(nemail);
        write('Please enter the remark: ');
        readln(nremark);
        n := num +1;
        for i := num downto 1 do
        if name[i] > nname
        then n := n-1;
        for i := num downto n do
        begin
                name[i+1] := name[i];
                sex[i+1] := sex[i];
                phone[i+1] := phone[i];
```

```
                email[i+1] := email[i];
                remark[i+1] := remark[i];
        end;
        name[n]:= nname;
        sex[n]:= nsex;
        phone[n]:= nphone;
        email[n]:= nemail;
        remark[n]:= nremark;
        num := num+1;
end;

procedure searchdata;
var found : boolean;
      target : string;
      i : integer;
begin
        clrscr;
        write('Please enter the target name : ');
        readln(target);
        found:= false;
        for i := 1 to num do
                if name[i] = target
                then begin
                                clrscr;
                                writeln;
                                writeln;
                                writeln('Name                  Sex          Phone                Email
Remark' );
                                writeln('-------------------------------------------------------------------------------' );

writeln(name[i],'':15-length(name[i]),sex[i],'':8,phone[i],'':15-length(phone[i]),email[i],'':20-length(email[i]),
remark[i]);
                                found := true
                        end;
          if not found
          then writeln('No such person')
end;

procedure deletedata;
var i, m : integer;
      target : string;
begin
```

```
        clrscr;
        m := 0 ;
        writeln('Name                Sex         Phone               Email                       Remark' );
        writeln('-------------------------------------------------------------------------------' );
        for i := 1 to num do

writeln(name[i],'':15-length(name[i]),sex[i],'':8,phone[i],'':15-length(phone[i]),email[i],'':20-length(email[i]),
remark[i]);
        writeln;
        writeln;
        write('Please enter the name of the target: ');
        readln(target);
        for i := 1 to num do
        if name[i] = target
        then begin
                        m := i;
                        if m > 0
                        then begin
                                        for i := m to num-1 do
                                        begin
                                                name[i] := name[i+1];
                                                sex[i] := sex[i+1];
                                                phone[i] := phone[i+1];
                                                email[i] := email[i+1];
                                                remark[i] := remark[i+1]
                                        end;
                                        writeln('Deleted');
                                        num := num-1;
                end
          else writeln('No such person')
          end;
end;

procedure changedata;
var i,choice : integer;
       target, phonetemp : string;
       found : boolean;
begin
        clrscr;
        writeln('Name                Sex         Phone               Email                       Remark' );
        writeln('-------------------------------------------------------------------------------' );
        for i := 1 to num do
```

```
writeln(name[i],'':15-length(name[i]),sex[i],'':8,phone[i],'':15-length(phone[i]),email[i],'':20-length(email[i]),
remark[i]);
        writeln;
        writeln;
        write('Please enter the name of the target: ');
        readln(target);
        found := false;
        for i := 1 to num do
        if name[i] = target
        then begin
                        clrscr;
                        writeln;
                        writeln;
                        writeln('Name                    Sex          Phone                  Email
Remark' );
                        writeln('-------------------------------------------------------------------------------' );

writeln(name[i],'':15-length(name[i]),sex[i],'':8,phone[i],'':15-length(phone[i]),email[i],'':20-length(email[i]),
remark[i]);
                        writeln;
                        writeln;
                        writeln(' 1 : Name');
                        writeln(' 2 : Sex');
                        writeln(' 3 : Phone');
                        writeln(' 4 : Email');
                        writeln(' 5 : Remark');
                        writeln;
                        write('What do you want to change? ');
                        readln(choice);
                        case choice of
                        1 : begin
                                        write('Please enter the new name: ');
                                        readln(name[i]);
                                        writeln('Changed');
                                        found := true;
                              end;
                        2 : begin
                                        write('Please enter the new sex: ');
                                        readln(sex[i]);
                                        writeln('Changed');
                                        found := true;
```

```
                        end;
                    3 : begin
                            write('Please enter the new phone: ');
                            readln(phonetemp);
                            if length(phonetemp)>8
                            then repeat
                                writeln('Wrong phone number');
                                write('Please enter again: ');
                                readln(phonetemp);
                            until (length(phonetemp)<=8);
                            phone[i] := phonetemp;
                            writeln('Changed');
                            found := true;
                        end;
                    4 : begin
                            write('Please enter the new email: ');
                            readln(email[i]);
                            writeln('Changed');
                            found := true;
                        end;
                    5 : begin
                            write('Please enter the new remark: ');
                            readln(remark[i]);
                            writeln('Changed');
                            found := true;
                        end;
                    end;
                end;
        if not found
        then writeln('No such person')
end;

procedure displaydata;
var i : integer;
begin
        clrscr;
        writeln('Name                Sex         Phone               Email                       Remark' );
        writeln('-------------------------------------------------------------------------------' );
        for i := 1 to num do
        begin

writeln(name[i],'':15-length(name[i]),sex[i],'':8,phone[i],'':15-length(phone[i]),email[i],'':20-length(email[i]),
```

```
remark[i])
        end
end;

procedure sortdata;
var i,n : integer;
      nametemp, sextemp, phonetemp, emailtemp, remarktemp: string;
      finish: boolean;
begin
        n:=1;
        repeat
                finish := true;
                for i := 1 to num-n do
                if name[i]>name[i+1]
                then begin
                                nametemp := name[i];
                                name[i] := name[i+1];
                                name[i+1] := nametemp;
                                sextemp := sex[i];
                                sex[i] := sex[i+1];
                                sex[i+1] := sextemp;
                                phonetemp := phone[i];
                                phone[i] := phone[i+1];
                                phone[i+1] := phonetemp;
                                emailtemp := email[i];
                                email[i] := email[i+1];
                                email[i+1] := emailtemp;
                                remarktemp := remark[i];
                                remark[i] := remark[i+1];
                                remark[i+1] := remarktemp;
                                finish := false
                        end;
                n := n+1
         until finish = true
end;

begin
        readbook;
        clrscr;
        textcolor(11);
        writeln;
        writeln;
```

```
        writeln;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln(' WW                WW    EEEEEEEEE LLL           CCCCCCCC OOOOOOOOO MMMM
MMMM EEEEEEEEE');
        writeln(' WW                WW    EEE           LLL           CC           OO         OO MM MM MM MM
EEE');
        writeln('    WW             WW     EEEEEEEEE LLL           CC           OO         OO MM    MMM
MM EEEEEEEEE');
        writeln('    WW      WW     WW       EEE           LLL           CC           OO         OO MM
MM EEE');
        writeln('     WW WWWW WW        EEE           LLL           CC           OO         OO MM
MM EEE');
        writeln('       WWW    WWW          EEEEEEEEE LLLLLLLLL CCCCCCCC OOOOOOOOO MM           MM
EEEEEEEEE');
        writeln;
        writeln;
        writeln;
        writeln('Please press Enter to continous');
        readln;
        repeat
        clrscr;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln('                                        My Phone Book');
        writeln('                  ================================================');
        writeln('                  ==              1 : Insert a New Data                    ==');
        writeln('                  ==              2 : Search the Data                       ==');
        writeln('                  ==              3 : Change the Data                       ==');
        writeln('                  ==              4 : Delete the Data                       ==');
        writeln('                  ==              5 : Sort the Data                         ==');
        writeln('                  ==              6 : Display All the Data                  ==');
        writeln('                  ==              7 : Store the Data                        ==');
        writeln('                  ==              8 : Close the Phone Book                  ==');
        writeln('                  ================================================');
        writeln;
        write('                           Please Enter the Number: ');
```

```
        readln(n);
        writeln;
                case n of
                1 : adddata;
                2 : searchdata;
                3 : changedata;
                4 : deletedata;
                5 : sortdata;
                6 : displaydata;
                7 : writebook
                end;
        readln;
        until n = 8;
    writebook
end.
```