Hong Kong Diploma of Secondary Education
Examination 201x
Information and Communication Technology

Option D: Software Development

Title: Phonebook system

Contents

Chapter 1 Introduction

1.1 Background

Mr. Chan is a very busy business man, he needs to contact all sort of people due to requirements of his job, such as clients, colleges, boss, and even friends. In the past, he uses a actual book to note down all the contacts, but he always felt that this way of noting down contacts is very inconvenient, due to several reason. Firstly, Mr. Chan is a very careless person, he always left his belonging everywhere, which makes his contacts extremely insecure, especially when the data stored in his phonebook is very valuable and highly confidential from clients.

Also, he finds that it takes a long time to find a specific contact in a huge amount of contact, which often affect his job due to the inconvenient, such as having to make his client wait, etc. Furthermore, he finds that his phonebook is very messy due to not being able to sort out contacts from different categories, it makes Mr. Chan having to spend a lot of time every time he wants to search for contact.

Seeing all this problem, he decided not to continue using such old-style way, and switch to use a computer phonebook program. He approached his younger brother, who is me, and asked me to create him a phonebook program.

1.2 Objectives

In this project, I aims to create a phonebook program specifically fits Mr. Chan's requirements. Mr. Chan has a couple of requirement when coming to the program, first, he must be able to find people in different group quickly, which also means that he need to be able to sort out contacts from different groups, also, he needs to be able to add notes to the contacts, due to the fact that there will be a lot of same name from different clients.

Based on the requirement of Mr. Chan, I have designed a phonebook program, which contains the following function:

1) Add a new contact and information about the contact

2) Edit a contact and its information

3) Delete a contact

4) To view the list of contact sorted by different method

The contact data will be stored at a separated text file, which will be edited every time the user is done editing.

Chapter 2 Design of Solution

2.1 Brief Description

This chapter will show the detail of designing the program, and explanation of different function of the program. The program aims to provide a modern way for Mr. Chan to store contacts, either for his job, or his friends and relatives. The contacts' data will be stored in a text file(txt), which the user is always able to backup. Each contact will contain the data of one's name, gender(limited to m or f), phone number(limited to numbers), email, and a description limited to 30 words.

When the user first enter the program, one will be greeted by a main menu, which consist of a total of five functions: add contact, view and edit contact, delete contact, search contact and exit program. each of them have a index number respectively. every time the user goes back to the main menu, the data will be transferred and updated to the external text file, stored in the same file with the program's root location.

2.2 Refinement of Problem

The phonebook program will need to perform a series of functions, including:

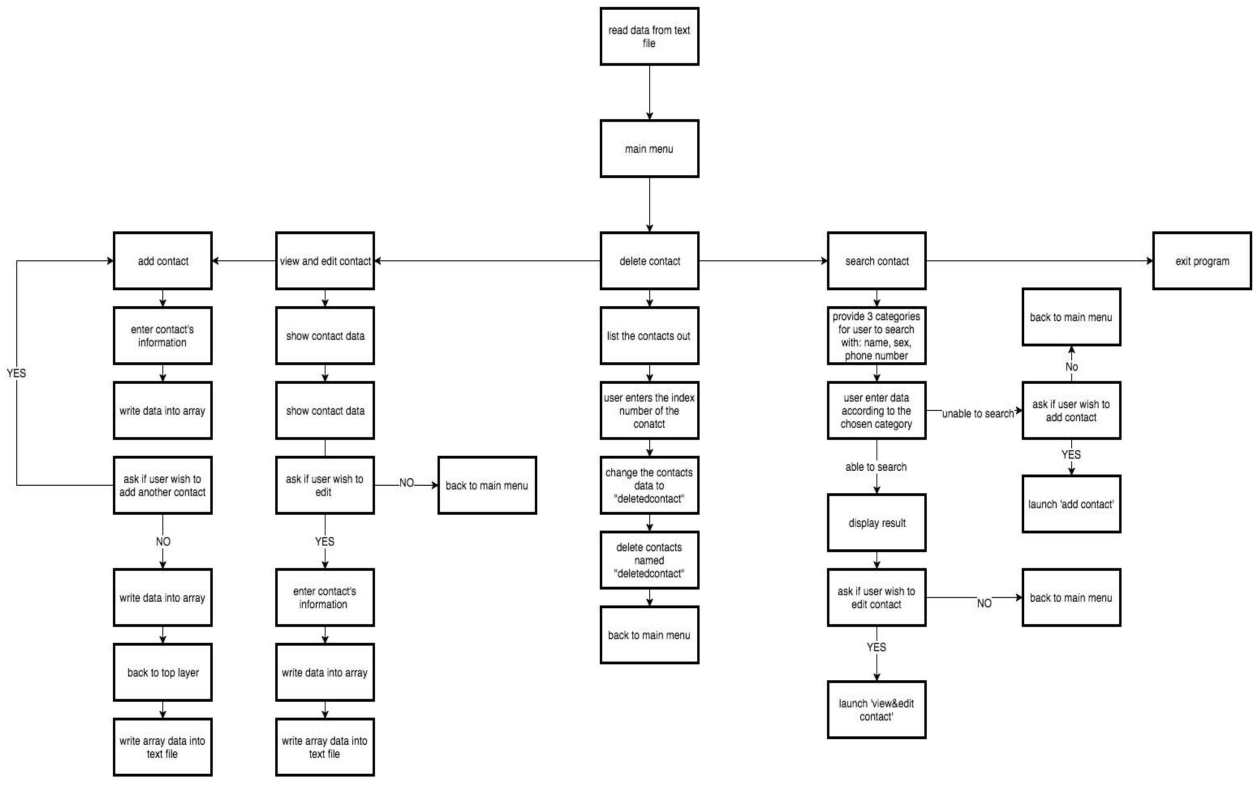
1) Adding a contact

2) Editing the data of a contact

3) Deleting a contact

4) Viewing a contact by different criteria

The above mentioned functions can be accessed through the main menu.


read data from text file
main menu
add contact
view and edit contact
delete contact
search contact
exit program
enter contact's information
write data into array
ask if user wish to add another contact
YES
NO
write data into array
back to top layer
write array data into text file
show contact data
show contact data
ask if user wish to edit
NO
back to main menu
YES
enter contact's information
write data into array
write array data into text file
list the contacts out
user enters the index number of the conatct
change the contacts data to "deletedcontact"
delete contacts named "deletedcontact"
back to main menu
provide 3 categories for user to search with: name, sex, phone number
user enter data according to the chosen category
unable to search
ask if user wish to add contact
No
back to main menu
YES
launch 'add contact'
able to search
display result
ask if user wish to edit contact
NO
back to main menu
YES
launch 'view&edit contact'

2.3 Input Data File Formats

The user will input various types of data of the contact, including name, phone number, gender and email address of the contact. All of the above mentioned data will be stored in a external file called ‘contact.txt’. The detail of each data available for input is shown as below:

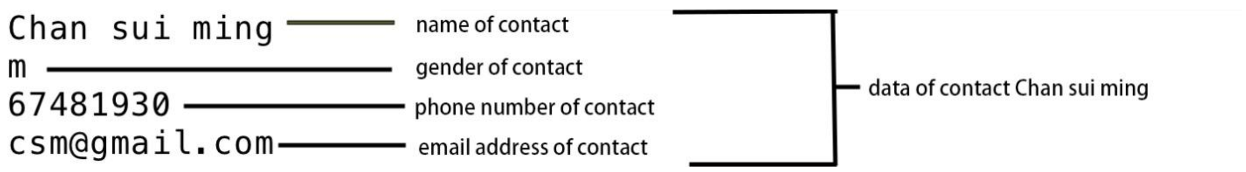
Name: a type of string not exceeding 16 characters.

Phone number : a type of string not exceeding 8 characters

Gender: a character, most preferably M and F

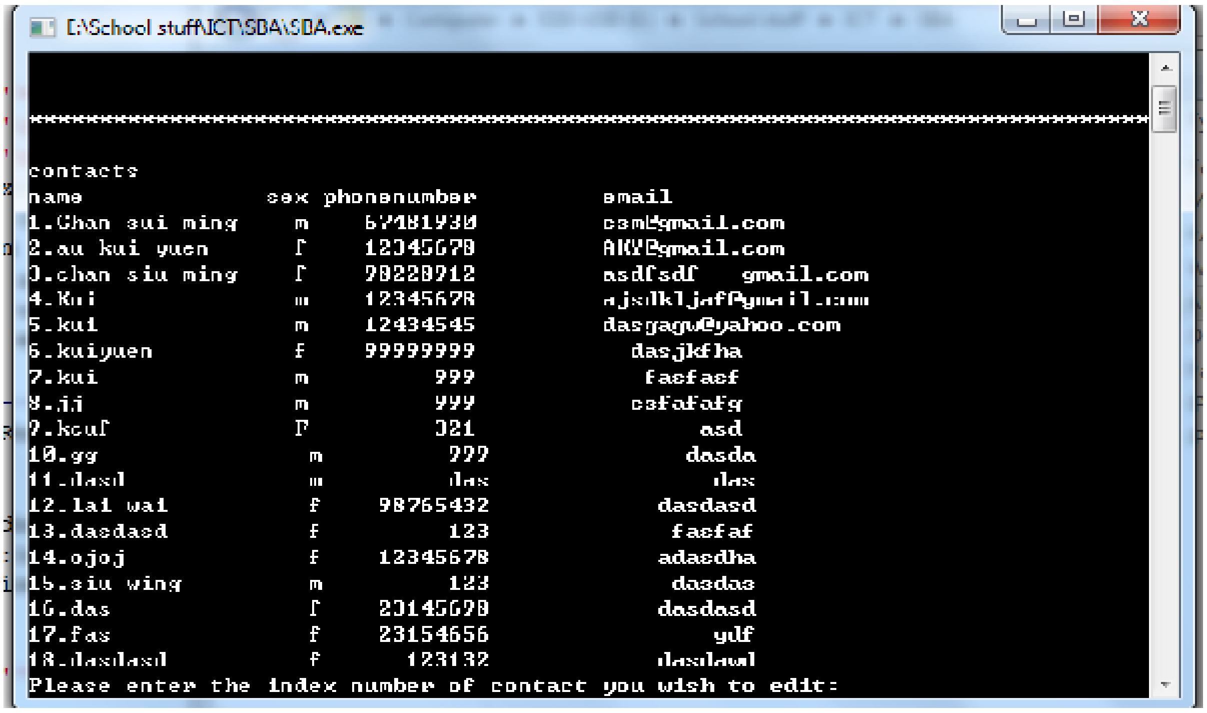
Email address: a type of string not exceeding 30 characters

In the external file storing the data of contacts, the data stored will be in the form of 4 column per contacts, stored in the following order:


```
Chan sui ming
m
67481930
csm@gmail.com
```

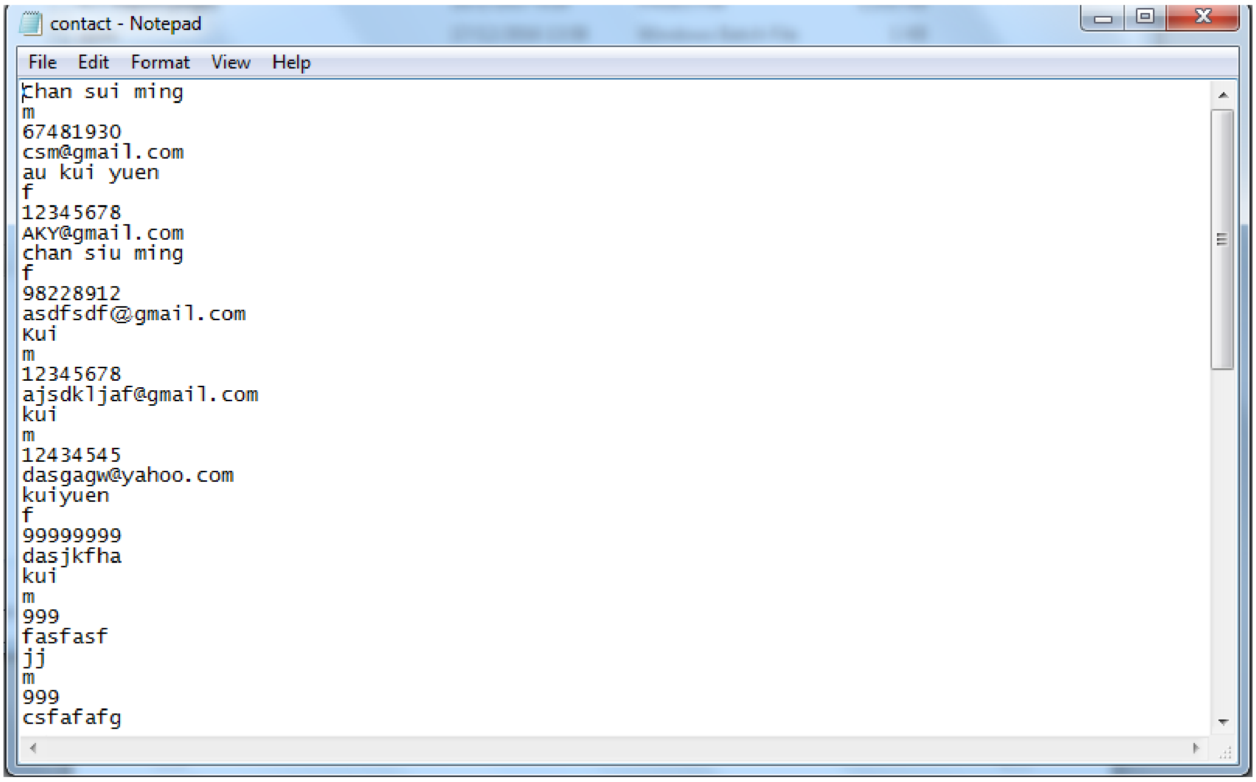
2.4 Output Report Format

In the function: view&edit contact and delete contact function, the user will be able to see the list of the contact, the way of displaying contact datas with be as follow:



```
********************************************************************************

contacts
name               sex phonenumber         email
1.Chan sui ming     m     67181930         csm@gmail.com
2.au kui yuen       f     12345678         AKY@gmail.com
3.chan siu ming     f     78228712         asdfsdf   gmail.com
4.Kui               m     12345678         ajsdlkljaf@gmail.com
5.kui               m     12434545         dasgagw@yahoo.com
6.kuiyuen           f     99999999           dasjkfha
7.kui               m          999            fasfasf
8.ji                m          999           csfafafg
9.kcuf              f          321                asd
10.gg                m          777              dasda
11.dasd              m          das                das
12.lai wai           f     98765432            dasdasd
13.dasdasd           f          123             fasfaf
14.ojoj              f     12345678            adasdha
15.siu wing          m          123             dasdas
16.das               f     23145678            dasdasd
17.fas               f     23154656                ydf
18.dasdasd           f       123132            dasdawd
Please enter the index number of contact you wish to edit:
```

The first column will be responsible for displaying the name contact, the second one will be for gender, third one is desgined for the phone number of the contact, finally the email will be shown in the last column.



```
chan sui ming
m
67481930
csm@gmail.com
au kui yuen
f
12345678
AKY@gmail.com
chan siu ming
f
98228912
asdfsdf@gmail.com
Kui
m
12345678
ajsdkljaf@gmail.com
kui
m
12434545
dasgagw@yahoo.com
kuiyuen
f
99999999
dasjkfha
kui
m
999
fasfasf
jj
m
999
csfafafg
```

The contact data will be stored in the following format, four row per contact.

Chapter 3 Implementation

3.1 Brief Description

In this chapter, i will briefly describe the implementation of the phone book program. in the following chapter, I will:

determine the data structure to be used in the program

describe the functions to be performed by each procedure in the program

explain the main algorithm to be used in the program

display some example of the program codes

display the user interface and analysis report

3.2 Data Structures

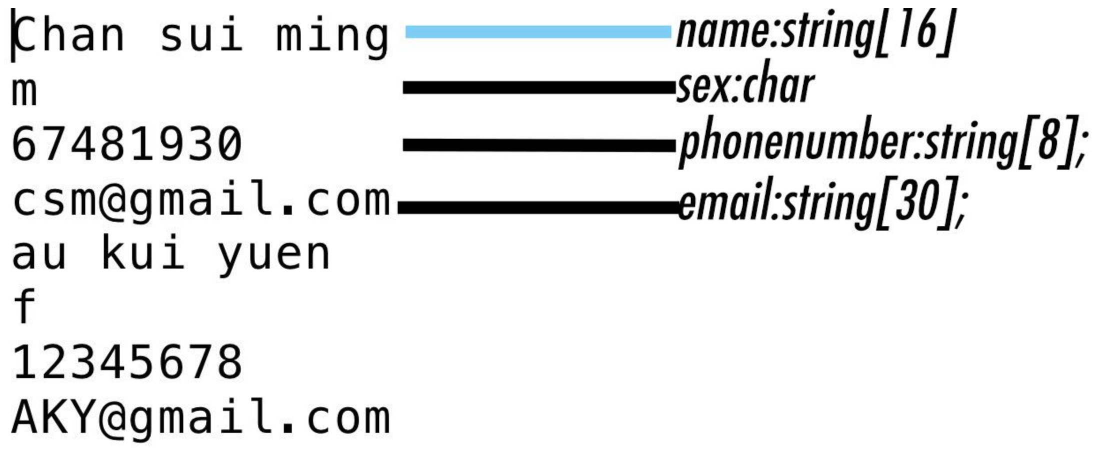
a data type called ‘contacttype' will be created, used to define the data stored in the contact file. It contains the following:

name: string[16]: name defined as a array not over 16 strings

sex: char: gender defined as a character

phonenumber: string[8]: phone number defined as a array not over 8 strings

email:string[30]; email defined as a array not over 30 strings



```
Chan sui ming
m
67481930
csm@gmail.com
au kui yuen
f
12345678
AKY@gmail.com
```

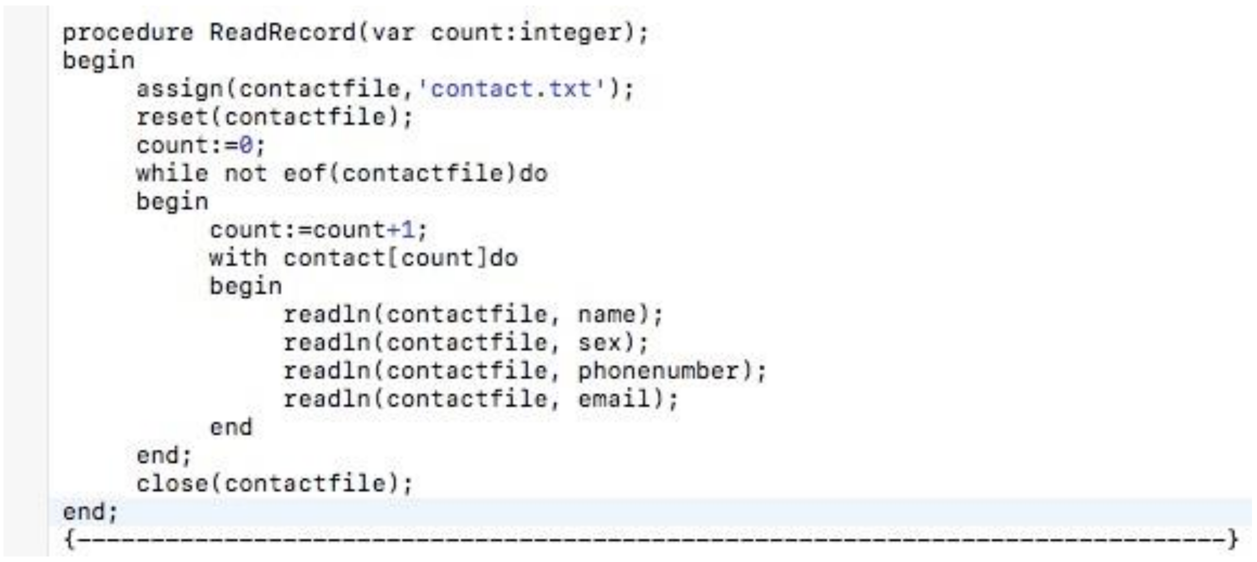
Contact data stored in the contact file will be as the form of a one-dimensional array, divided by four row per contact.

3.3 Procedures in the Program

A total of 7 procedures is used in the program, they are all written according to the system chart on page 5, here is the procedures:

A) Procedure read record

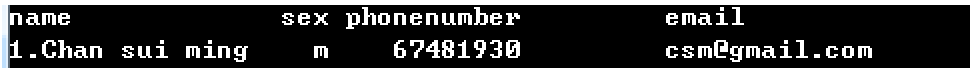
This procedure is responsible for reading data of the contacts in the external text file, which should be named ‘contact.txt’. Also, this procedure will assign ‘contact.txt’ as the ‘contactfile’ used throughout the entire program. It performs the above mentioned functions by using the assign function and by reading through contact datas line by line, writing read data in to the array for the program to use. The program code are as below.



```
procedure ReadRecord(var count:integer);
begin
     assign(contactfile,'contact.txt');
     reset(contactfile);
     count:=0;
     while not eof(contactfile)do
     begin
          count:=count+1;
          with contact[count]do
          begin
               readln(contactfile, name);
               readln(contactfile, sex);
               readln(contactfile, phonenumber);
               readln(contactfile, email);
          end
     end;
     close(contactfile);
end;
{------------------------------------------------------------------------------}
```

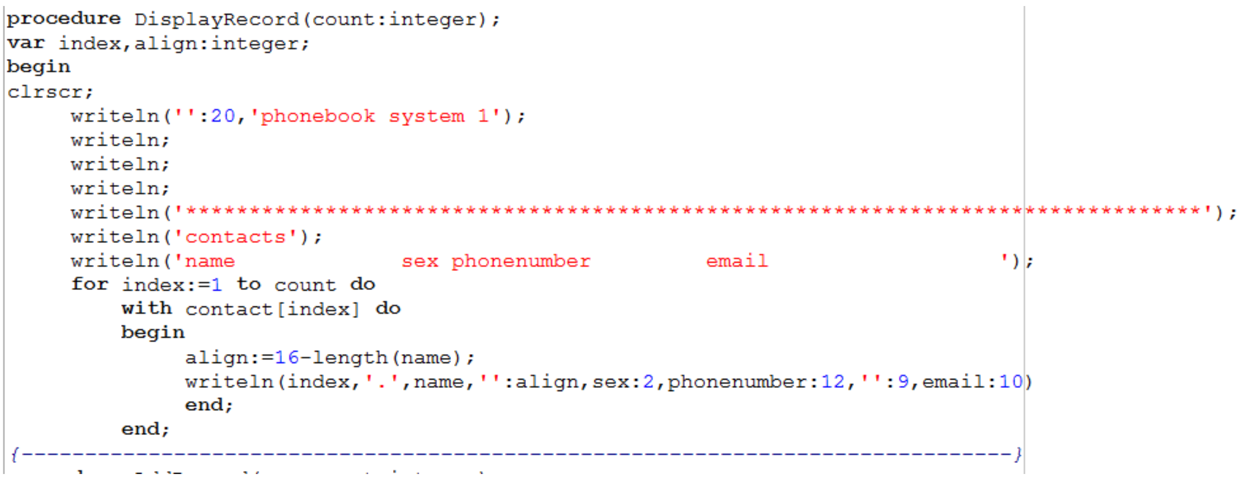
B) procedure displayrecord

This procedure is designed to be used in various main functions, which requires the contact info to be shown, the procedure will display the contact's info in the form of following:



```
name                 sex phonenumber          email
1.Chan sui ming        m    67481930          csm@gmail.com
```

The contact's info will be aligned in the form of index number, name, gender, phone number and email, as shown above

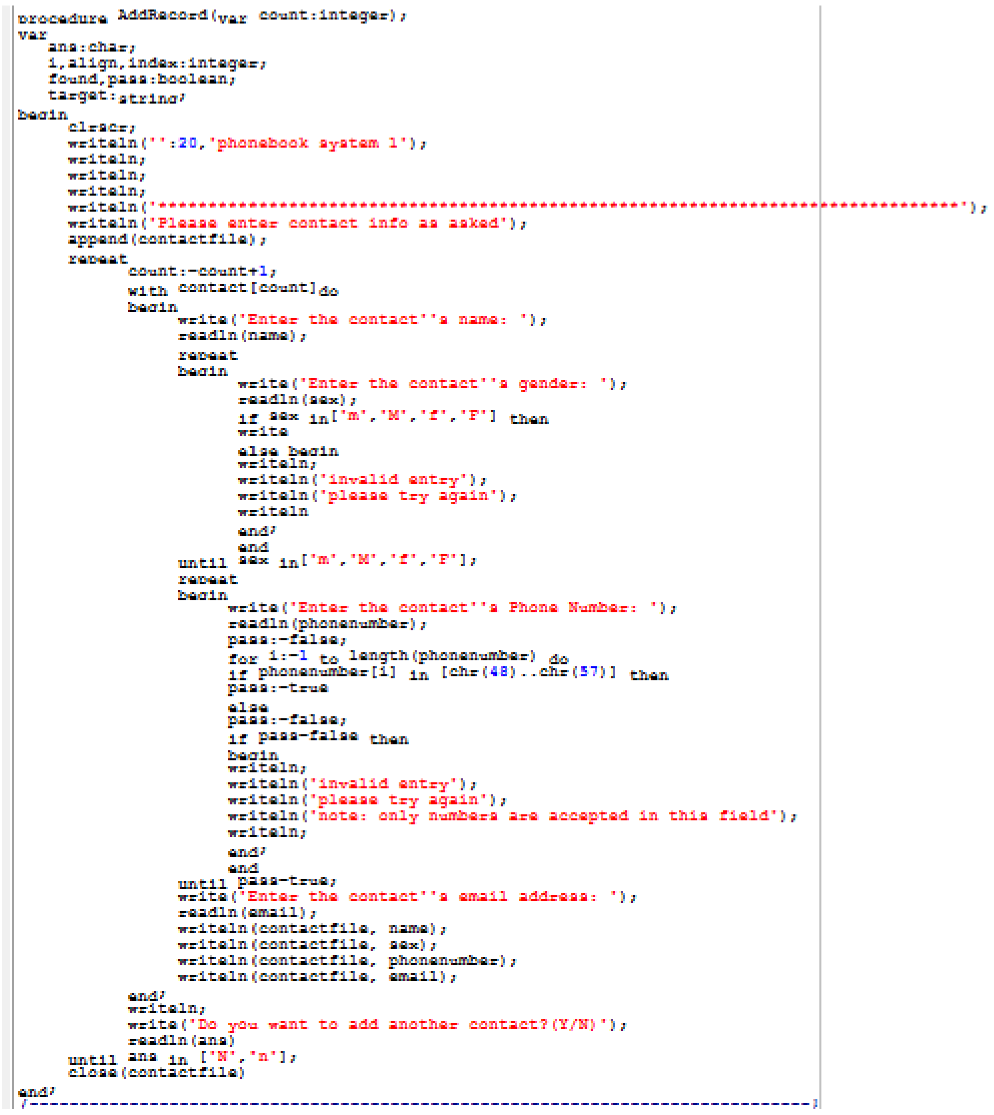


```
procedure DisplayRecord(count:integer);
var index,align:integer;
begin
clrscr;
     writeln('':20,'phonebook system 1');
     writeln;
     writeln;
     writeln;
     writeln('********************************************************************************');
     writeln('contacts');
     writeln('name            sex phonenumber         email                  ');
     for index:=1 to count do
         with contact[index] do
         begin
              align:=16-length(name);
              writeln(index,'.',name,'':align,sex:2,phonenumber:12,'':9,email:10)
              end;
         end;
{-------------------------------------------------------------------------------}
```

c) procedure add record

This procedure will be responsible for any action by the user involving adding data in the contact file, this is one of the four main function that will be shown in the main menu. Once the user select the add record function, the system will guide the user to input a series of data including name, gender(limited to M, m, F, f here), phone number(limited to numbers here), and email. After performing the above mentioned action, the system will ask the user if he wish to edit again, if not, the system will return to the main menu

Here is the screenshot of the procedure:

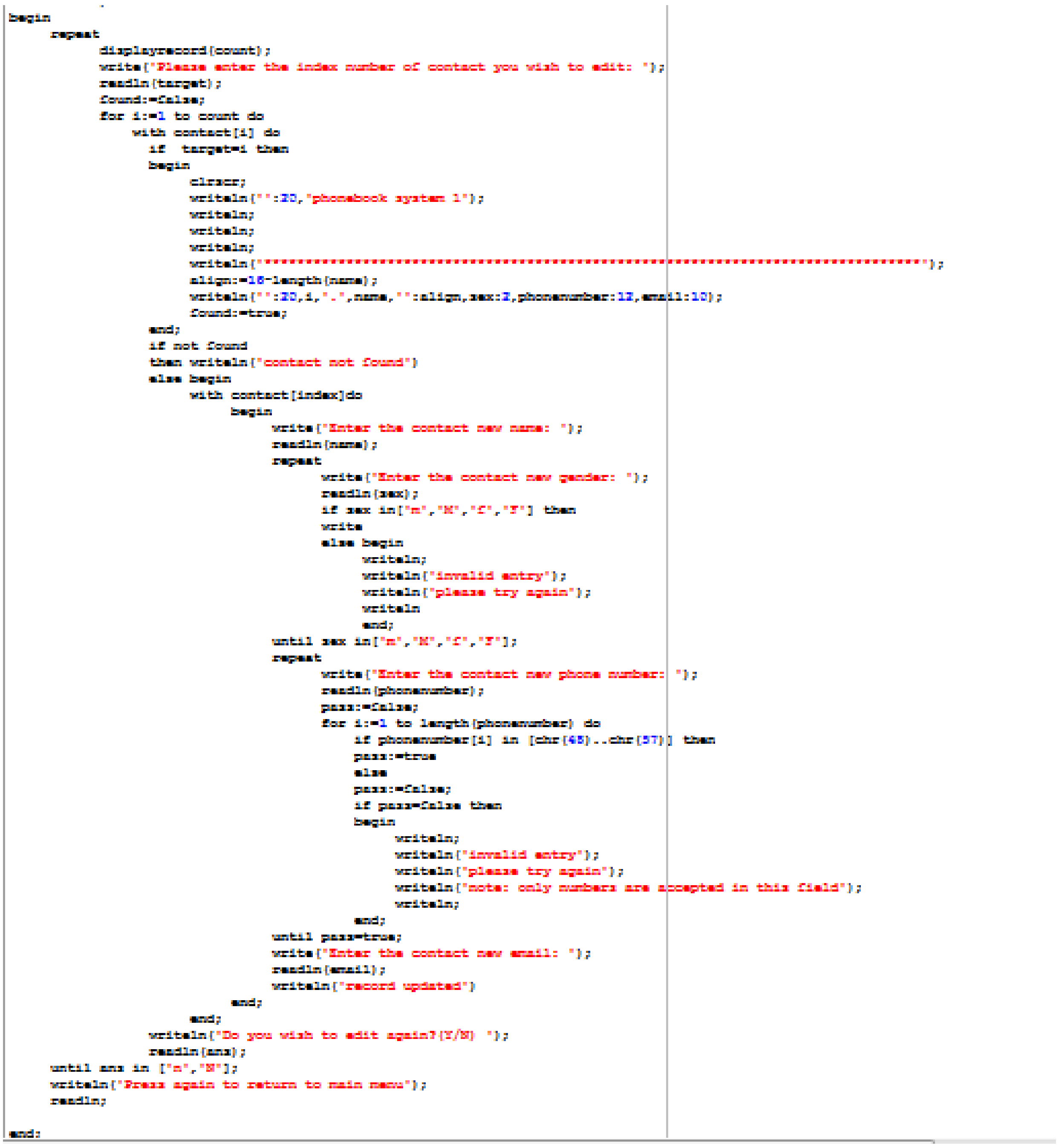


```
procedure AddRecord(var count:integer);
var
    ans:char;
    i,align,index:integer;
    found,pass:boolean;
    target:string;
begin
      clrscr;
      writeln('':20,'phonebook system 1');
      writeln;
      writeln;
      writeln;
      writeln('********************************************************************************');
      writeln('Please enter contact info as asked');
      append(contactfile);
      repeat
              count:=count+1;
              with contact[count]do
              begin
                    write('Enter the contact''s name: ');
                    readln(name);
                    repeat
                    begin
                            write('Enter the contact''s gender: ');
                            readln(sex);
                            if sex in['m','M','f','F'] then
                            write
                            else begin
                            writeln;
                            writeln('invalid entry');
                            writeln('please try again');
                            writeln
                            end;
                            end
                    until sex in['m','M','f','F'];
                    repeat
                    begin
                           write('Enter the contact''s Phone Number: ');
                           readln(phonenumber);
                           pass:=false;
                           for i:=1 to length(phonenumber) do
                           if phonenumber[i] in [chr(48)..chr(57)] then
                           pass:=true
                           else
                           pass:=false;
                           if pass=false then
                           begin
                           writeln;
                           writeln('invalid entry');
                           writeln('please try again');
                           writeln('note: only numbers are accepted in this field');
                           writeln;
                           end;
                           end
                    until pass=true;
                    write('Enter the contact''s email address: ');
                    readln(email);
                    writeln(contactfile, name);
                    writeln(contactfile, sex);
                    writeln(contactfile, phonenumber);
                    writeln(contactfile, email);
              end;
              writeln;
              write('Do you want to add another contact?(Y/N)');
              readln(ans)
      until ans in ['N','n'];
      close(contactfile)
end;
{--------------------------------------------------------------------------------}
```

D) view and edit contact

This particular procedure is responsible for editing the contact’s data. This procedure is the second of the main function displayed in the main menu, once the user selected this function, the system will display all the contact’s record in the program, each contact with a index number, the system will ask the user to enter the index number of the contact he wish to edit, then the procedure will clear the screen, and enter the second main function of the procedure, which is to edit the data of contact. The system will ask for the new data of the target contact, then the data will be updated once the user is back to the main menu.

Here is the screenshot of the procedure:

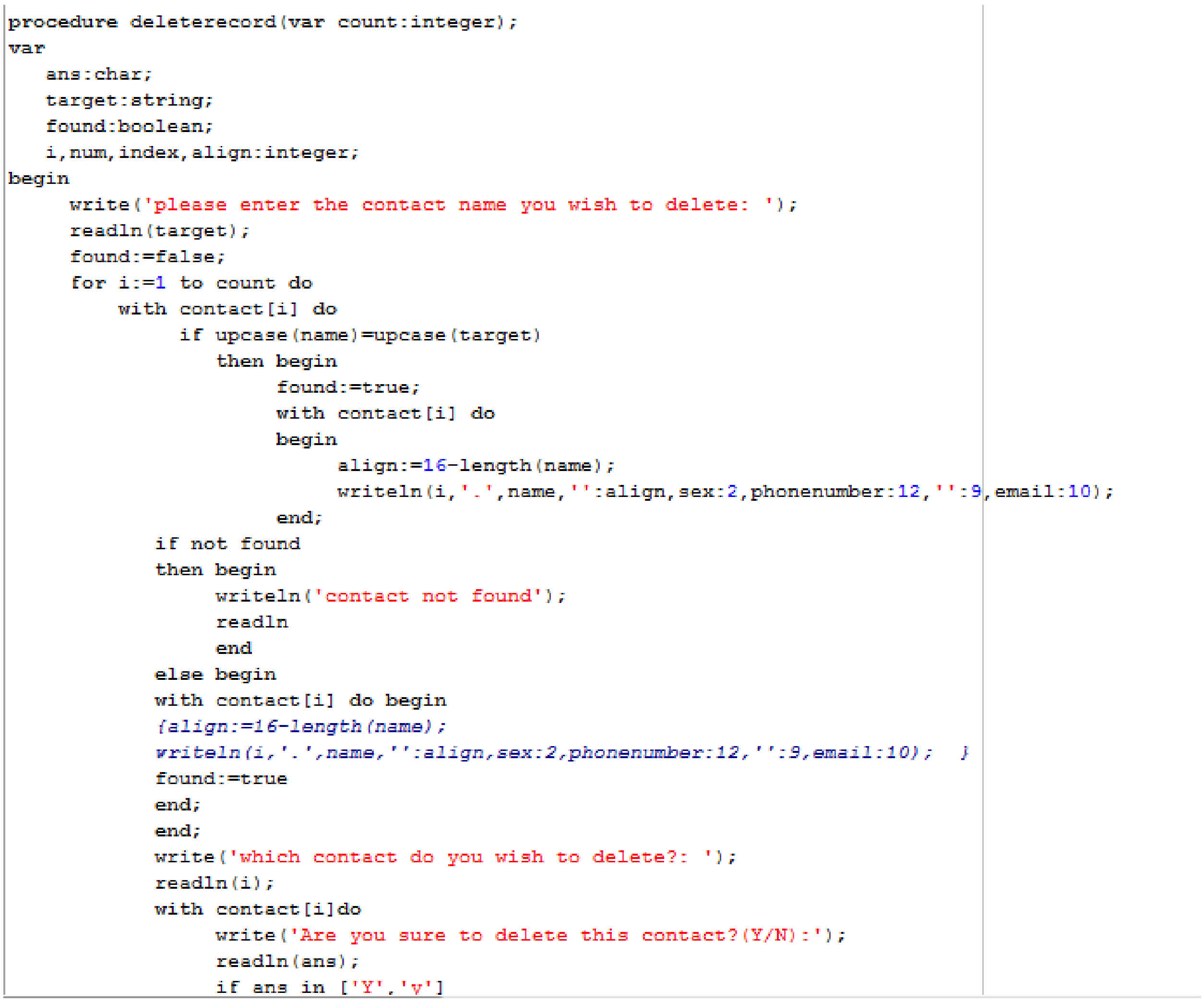


```
begin
    repeat
        displayrecord(count);
        write('Please enter the index number of contact you wish to edit: ');
        readln(target);
        found:=false;
        for i:=1 to count do
            with contact[i] do
             if  target=i then
             begin
                 clrscr;
                 writeln('':20,'phonebook system 1');
                 writeln;
                 writeln;
                 writeln;
                 writeln('********************************************************************************');
                 align:=18-length(name);
                 writeln('':20,i,'.',name,'':align,sex:2,phonenumber:12,email:10);
                 found:=true;
             end;
             if not found
             then writeln('contact not found')
             else begin
                 with contact[index]do
                     begin
                         write('Enter the contact new name: ');
                         readln(name);
                         repeat
                             write('Enter the contact new gender: ');
                             readln(sex);
                             if sex in['m','M','f','F'] then
                             write
                             else begin
                                 writeln;
                                 writeln('invalid entry');
                                 writeln('please try again');
                                 writeln
                                 end;
                         until sex in['m','M','f','F'];
                         repeat
                             write('Enter the contact new phone number: ');
                             readln(phonenumber);
                             pass:=false;
                             for i:=1 to length(phonenumber) do
                                if phonenumber[i] in [chr(48)..chr(57)] then
                                pass:=true
                                else
                                pass:=false;
                                if pass=false then
                                begin
                                    writeln;
                                    writeln('invalid entry');
                                    writeln('please try again');
                                    writeln('note: only numbers are accepted in this field');
                                    writeln;
                                end;
                         until pass=true;
                         write('Enter the contact new email: ');
                         readln(email);
                         writeln('record updated')
                     end;
                 end;
             writeln('Do you wish to edit again?(Y/N)  ');
             readln(ans);
    until ans in ['n','N'];
    writeln('Press again to return to main menu');
    readln;

end;
```

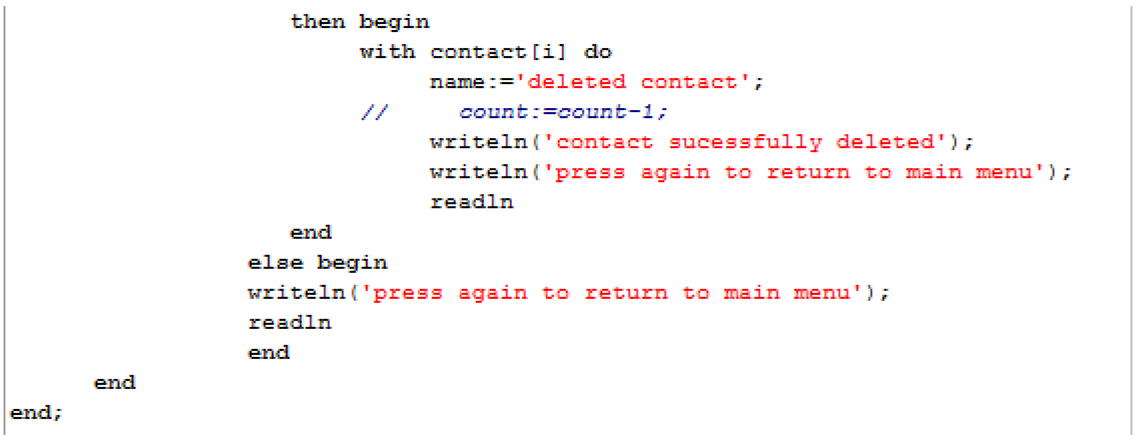
E) procedure delete record

This function is the third one of the four main function of the program, this function is responsible for deleting the unwanted contact in the contact file, once the user select this procedure, he will be greeted by the same contact data list as the view and edit contact. The user will enter the contact's index number, and after the confirmation process, the selected contact's data will be changed to 'deleted contact', and the next time the contact file is updated, it will not be written in to the contact file.

Here is the screenshot of the procedure:



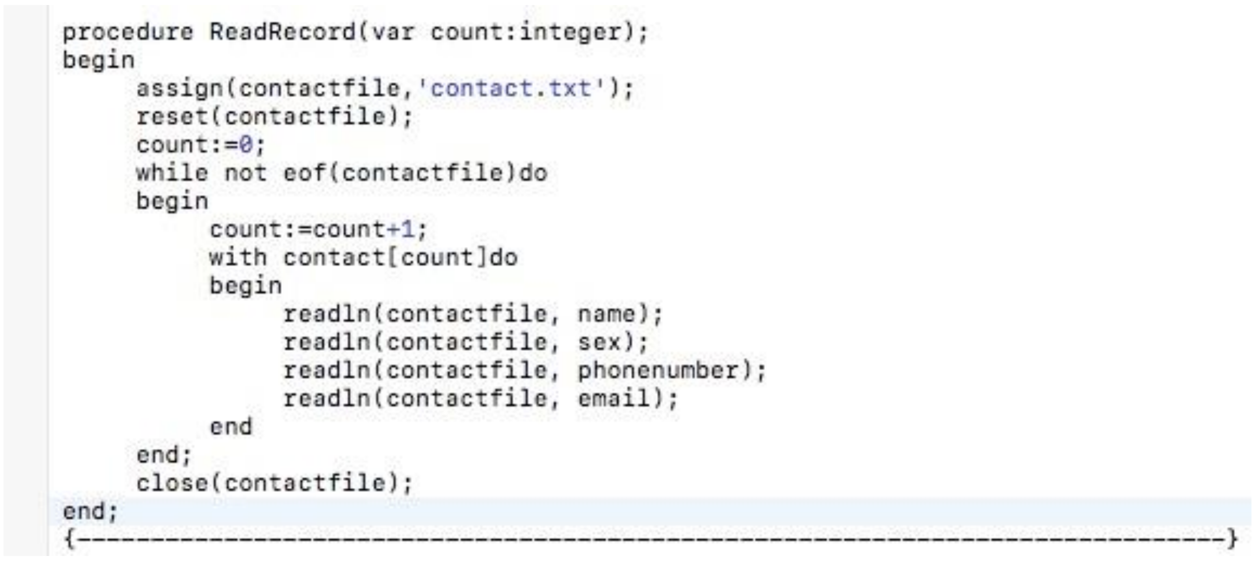
```
procedure deleterecord(var count:integer);
var
   ans:char;
   target:string;
   found:boolean;
   i,num,index,align:integer;
begin
     write('please enter the contact name you wish to delete: ');
     readln(target);
     found:=false;
     for i:=1 to count do
         with contact[i] do
              if upcase(name)=upcase(target)
                 then begin
                      found:=true;
                      with contact[i] do
                      begin
                           align:=16-length(name);
                           writeln(i,'.',name,'':align,sex:2,phonenumber:12,'':9,email:10);
                      end;
            if not found
            then begin
                 writeln('contact not found');
                 readln
                 end
            else begin
            with contact[i] do begin
            {align:=16-length(name);
            writeln(i,'.',name,'':align,sex:2,phonenumber:12,'':9,email:10);   }
            found:=true
            end;
            end;
            write('which contact do you wish to delete?: ');
            readln(i);
            with contact[i]do
                 write('Are you sure to delete this contact?(Y/N):');
                 readln(ans);
                 if ans in ['Y','y']
```



```
                    then begin
                         with contact[i] do
                              name:='deleted contact';
                         //     count:=count-1;
                              writeln('contact sucessfully deleted');
                              writeln('press again to return to main menu');
                              readln
                    end
                 else begin
                 writeln('press again to return to main menu');
                 readln
                 end
      end
end;
```

F) Procedure write record

This procedure is responsible for writing the data stored in the array of the program into the external text file, here is the screenshot of the procedure:

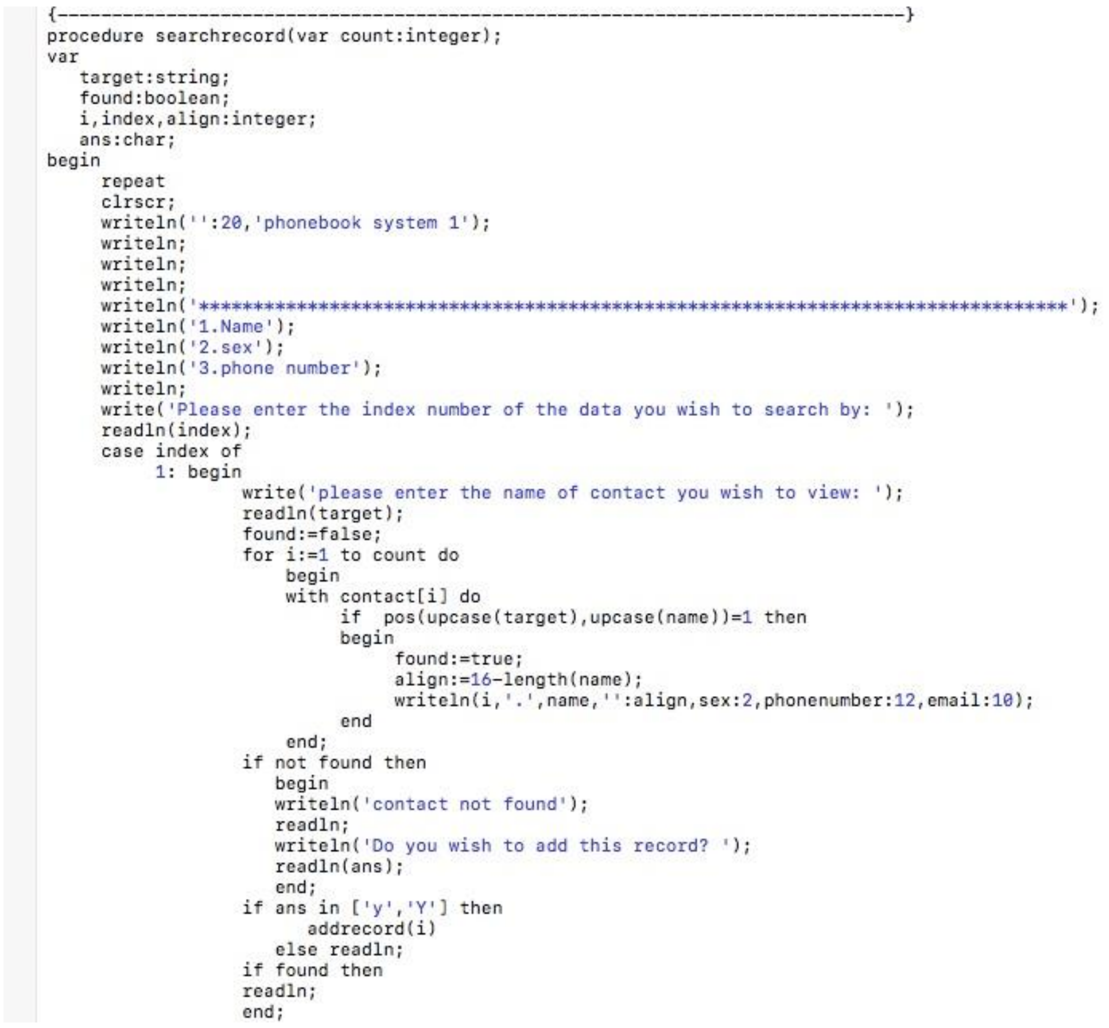


```
procedure ReadRecord(var count:integer);
begin
     assign(contactfile,'contact.txt');
     reset(contactfile);
     count:=0;
     while not eof(contactfile)do
     begin
          count:=count+1;
          with contact[count]do
          begin
               readln(contactfile, name);
               readln(contactfile, sex);
               readln(contactfile, phonenumber);
               readln(contactfile, email);
          end
     end;
     close(contactfile);
end;
{-------------------------------------------------------------------------------}
```

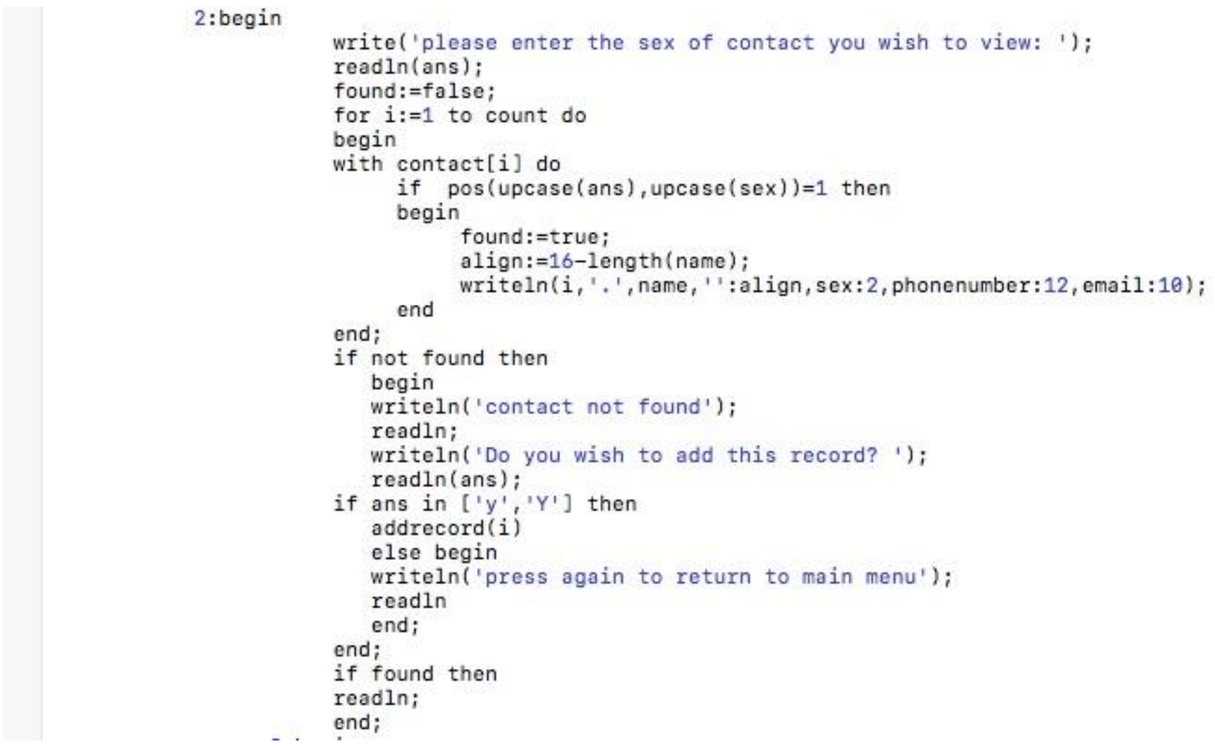
G) procedure search record

This procedure is the final one of the four main function displayed in the main menu, it is responsible for allowing the user to search for contact they wish, by entering some of the key information of the contact. When the user select this procedure in the main menu, three option will be shown on the screen, which is to search by name, gender, phone number respectively, by entering the selected information contacts with the matching data will be shown on the screen. if not found, the user will be given the option to add the contact immediately.

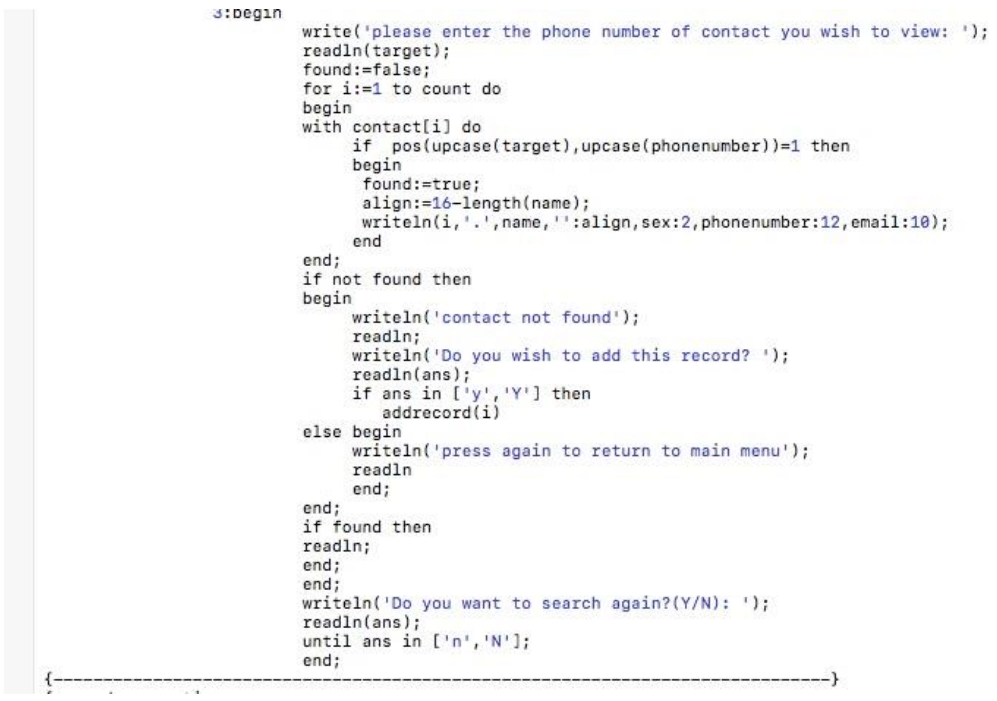
Here is the screenshot of the procedure:



```
{-------------------------------------------------------------------------------}
procedure searchrecord(var count:integer);
var
   target:string;
   found:boolean;
   i,index,align:integer;
   ans:char;
begin
     repeat
     clrscr;
     writeln('':20,'phonebook system 1');
     writeln;
     writeln;
     writeln;
     writeln('********************************************************************************');
     writeln('1.Name');
     writeln('2.sex');
     writeln('3.phone number');
     writeln;
     write('Please enter the index number of the data you wish to search by: ');
     readln(index);
     case index of
          1: begin
                    write('please enter the name of contact you wish to view: ');
                    readln(target);
                    found:=false;
                    for i:=1 to count do
                        begin
                        with contact[i] do
                             if  pos(upcase(target),upcase(name))=1 then
                             begin
                                  found:=true;
                                  align:=16-length(name);
                                  writeln(i,'.',name,'':align,sex:2,phonenumber:12,email:10);
                             end
                        end;
                    if not found then
                       begin
                       writeln('contact not found');
                       readln;
                       writeln('Do you wish to add this record? ');
                       readln(ans);
                       end;
                    if ans in ['y','Y'] then
                          addrecord(i)
                       else readln;
                    if found then
                    readln;
                    end;
```

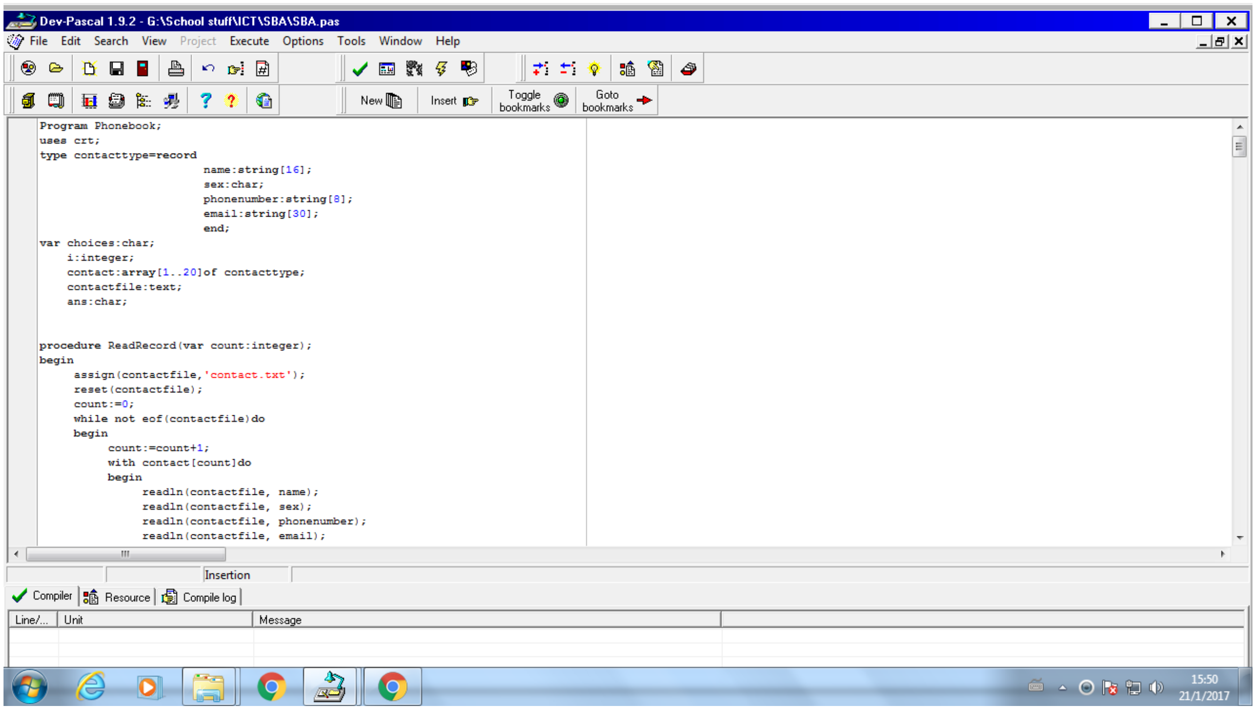


```pascal
2:begin
       write('please enter the sex of contact you wish to view: ');
       readln(ans);
       found:=false;
       for i:=1 to count do
       begin
       with contact[i] do
            if  pos(upcase(ans),upcase(sex))=1 then
            begin
                 found:=true;
                 align:=16-length(name);
                 writeln(i,'.',name,'':align,sex:2,phonenumber:12,email:10);
            end
       end;
       if not found then
          begin
          writeln('contact not found');
          readln;
          writeln('Do you wish to add this record? ');
          readln(ans);
       if ans in ['y','Y'] then
          addrecord(i)
          else begin
          writeln('press again to return to main menu');
          readln
          end;
       end;
       if found then
       readln;
       end;
```



```pascal
3:begin
        write('please enter the phone number of contact you wish to view: ');
        readln(target);
        found:=false;
        for i:=1 to count do
        begin
        with contact[i] do
             if  pos(upcase(target),upcase(phonenumber))=1 then
             begin
              found:=true;
              align:=16-length(name);
              writeln(i,'.',name,'':align,sex:2,phonenumber:12,email:10);
             end
        end;
        if not found then
        begin
             writeln('contact not found');
             readln;
             writeln('Do you wish to add this record? ');
             readln(ans);
             if ans in ['y','Y'] then
                addrecord(i)
        else begin
             writeln('press again to return to main menu');
             readln
             end;
        end;
        if found then
        readln;
        end;
        end;
        writeln('Do you want to search again?(Y/N): ');
        readln(ans);
        until ans in ['n','N'];
        end;
{-------------------------------------------------------------------------------}
```

3.4 Program Coding


Dev-Pascal 1.9.2 - G:\School stuff\ICT\SBA\SBA.pas
File Edit Search View Project Execute Options Tools Window Help
New
Insert
Toggle bookmarks
Goto bookmarks
Insertion
Compiler
Resource
Compile log
Line/...
Unit
Message
15:50
21/1/2017

```
Program Phonebook;
uses crt;
type contacttype=record
                        name:string[16];
                        sex:char;
                        phonenumber:string[8];
                        email:string[30];
                        end;
var choices:char;
    i:integer;
    contact:array[1..20]of contacttype;
    contactfile:text;
    ans:char;


procedure ReadRecord(var count:integer);
begin
     assign(contactfile,'contact.txt');
     reset(contactfile);
     count:=0;
     while not eof(contactfile)do
     begin
          count:=count+1;
          with contact[count]do
          begin
               readln(contactfile, name);
               readln(contactfile, sex);
               readln(contactfile, phonenumber);
               readln(contactfile, email);
```

This program is developed by pascal.

The program is named ‘phonebook.exe’, while the source code is named ‘phonebook.pas’.

The source code is placed in the appendix.

3.5 Program Execution

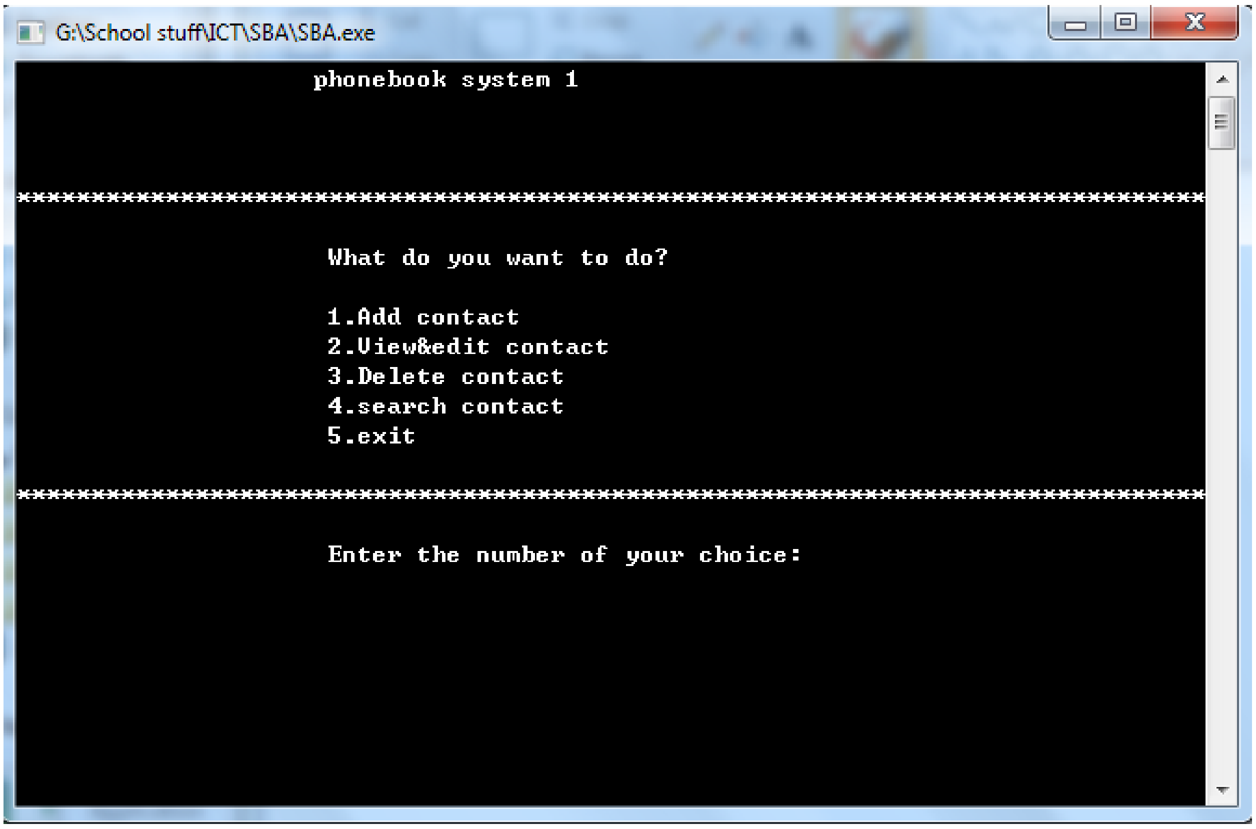
Requirement before execution:

A text file containing the data of the contacts named ‘contacts’ is required for launching the program. it must be placed in the same file with the program and the source code.

The program will be displayed in the window size of 80x25, the buffer size will be in the size of 80 width and 300 height, the window will be positioned by the system.

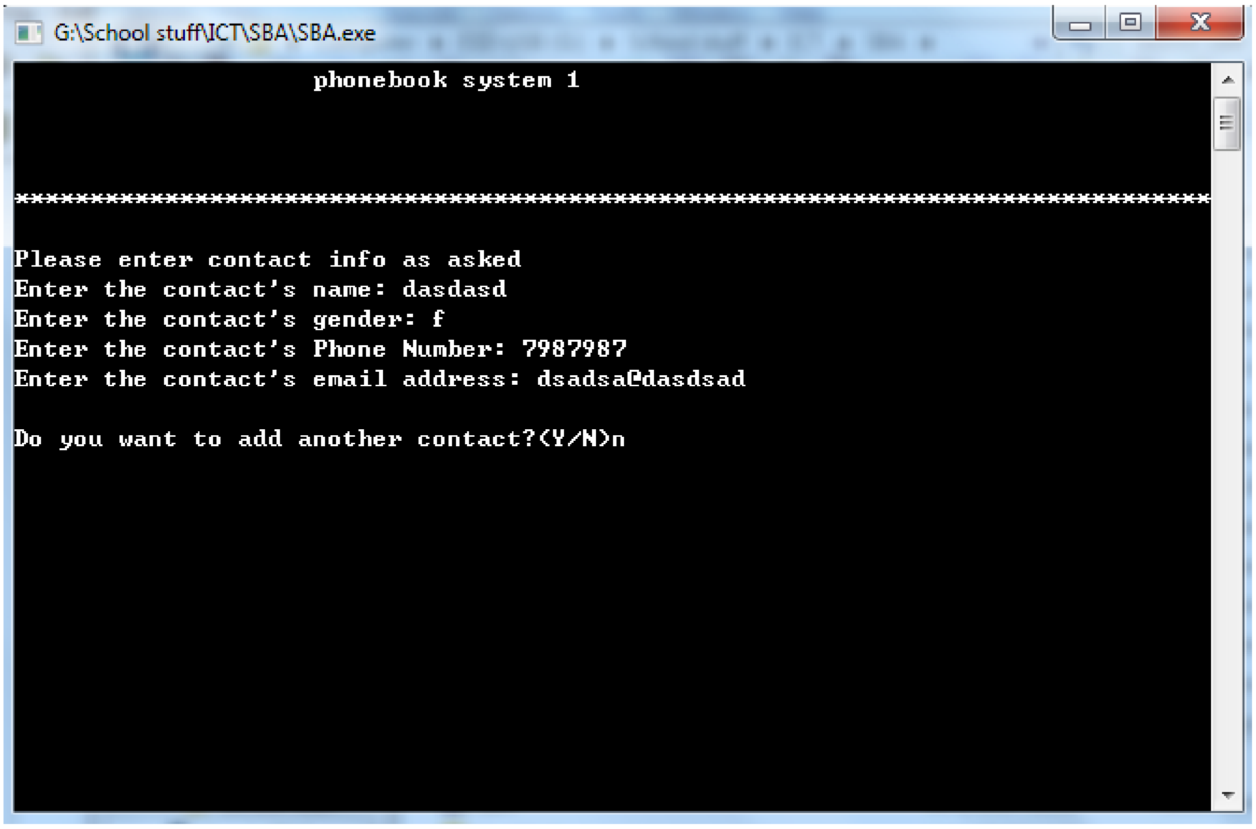
This specific phonebook is designed for one single user usage, no multi-user function is implemented.

User interface



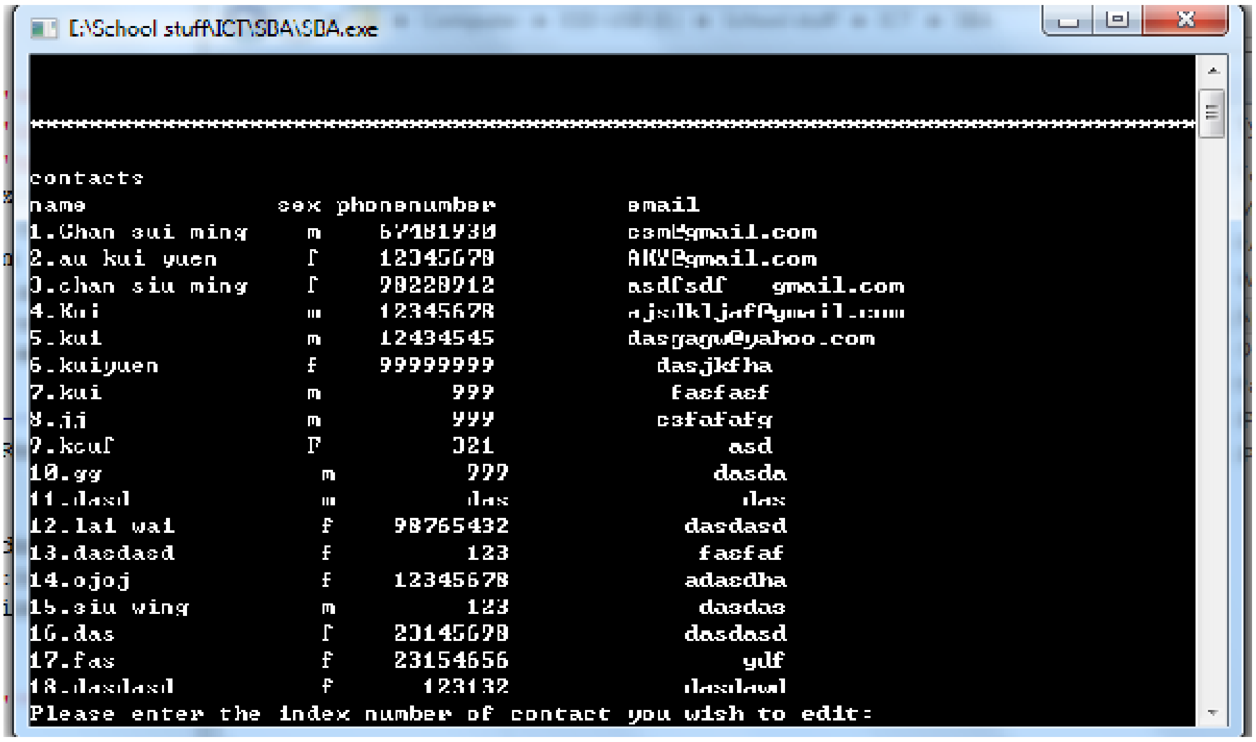
```
                    phonebook system 1



********************************************************************************

                     What do you want to do?

                     1.Add contact
                     2.View&edit contact
                     3.Delete contact
                     4.search contact
                     5.exit

********************************************************************************

                     Enter the number of your choice:
```

1)main menu


G:\School stuff\ICT\SBA\SBA.exe

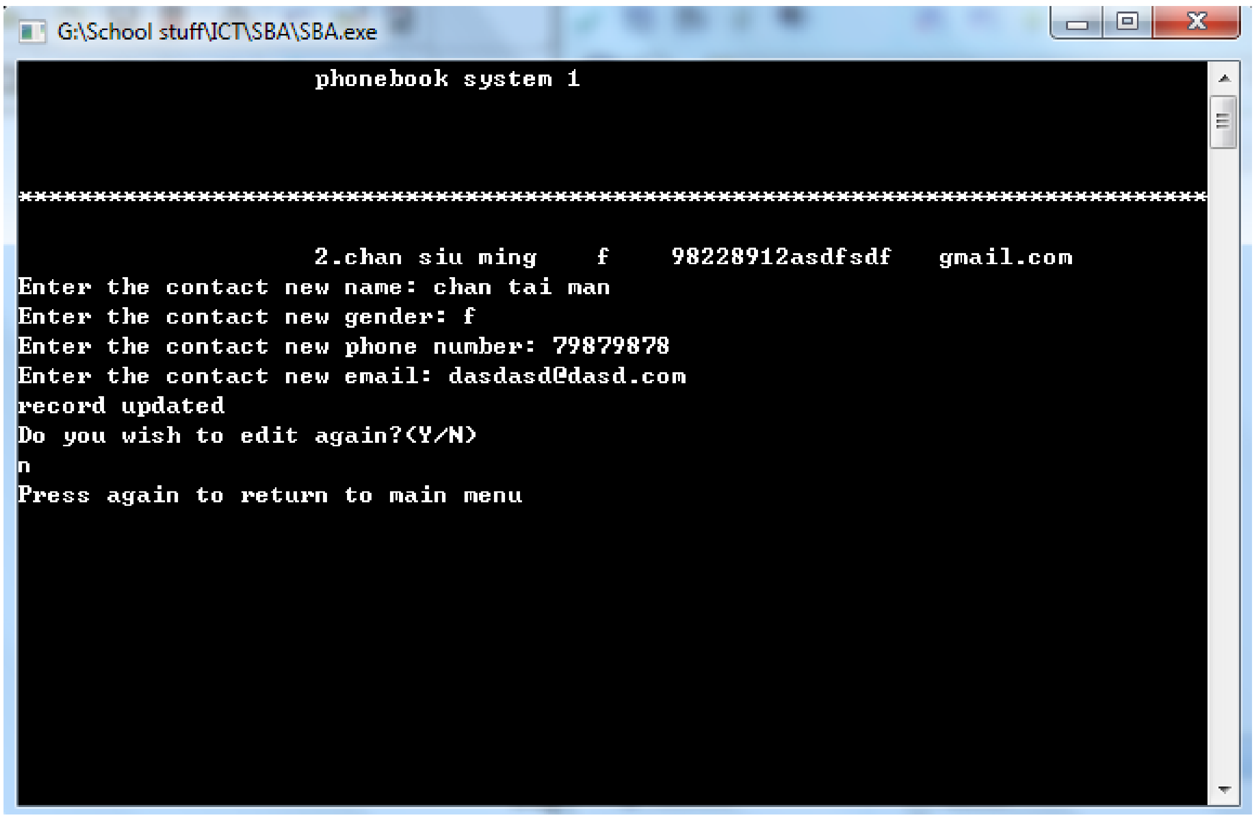
```
                          phonebook system 1



****************************************************************************************************

Please enter contact info as asked
Enter the contact's name: dasdasd
Enter the contact's gender: f
Enter the contact's Phone Number: 7987987
Enter the contact's email address: dsadsa@dasdsad

Do you want to add another contact?(Y/N)n
```

2)add record



```
E:\School stuff\ICT\SBA\SBA.exe

********************************************************************************

contacts
name             sex phonenumber          email
1.Chan sui ming    m     67181930          csm@gmail.com
2.au kui yuen      f     12345678          AKY@gmail.com
3.chan siu ming    f     78228912          asdfsdf    gmail.com
4.Kui              m     12345678          ajsdkljaf@gmail.com
5.kui              m     12434545          dasgagw@yahoo.com
6.kuiyuen          f     99999999            dasjkfha
7.kui              m          999             facfacf
8.ji               m          999            csfafafg
9.kcuf             F          321                 asd
10.gg               m          777               dasda
11.dasd             m          das                 das
12.lai wai          f     98765432             dasdasd
13.dasdasd          f          123              facfaf
14.ojoj             f     12345678             adacdha
15.siu wing         m          123              dasdas
16.das              f     23145678             dasdasd
17.fas              f     23154656                 ydf
18.dasdasd          f       123132             dasdawl
Please enter the index number of contact you wish to edit:
```

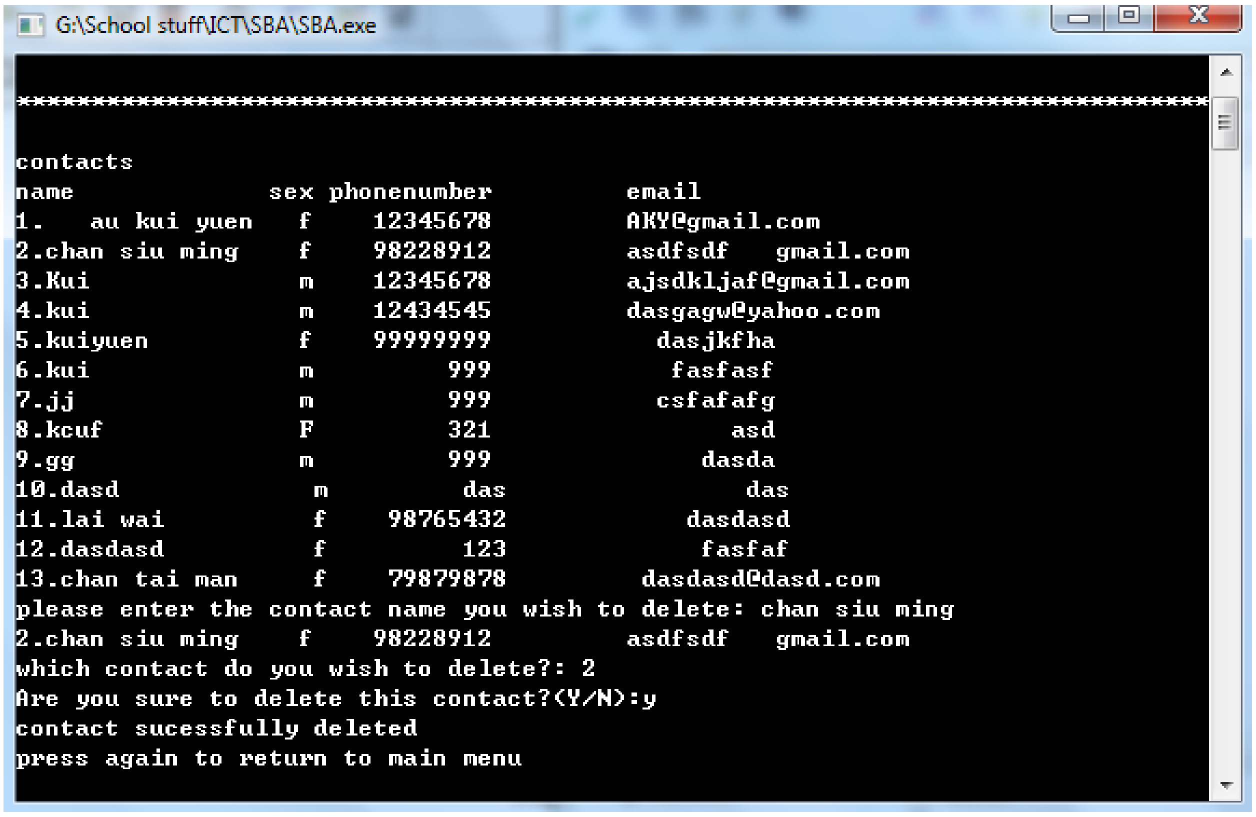
3)view record



G:\School stuff\ICT\SBA\SBA.exe

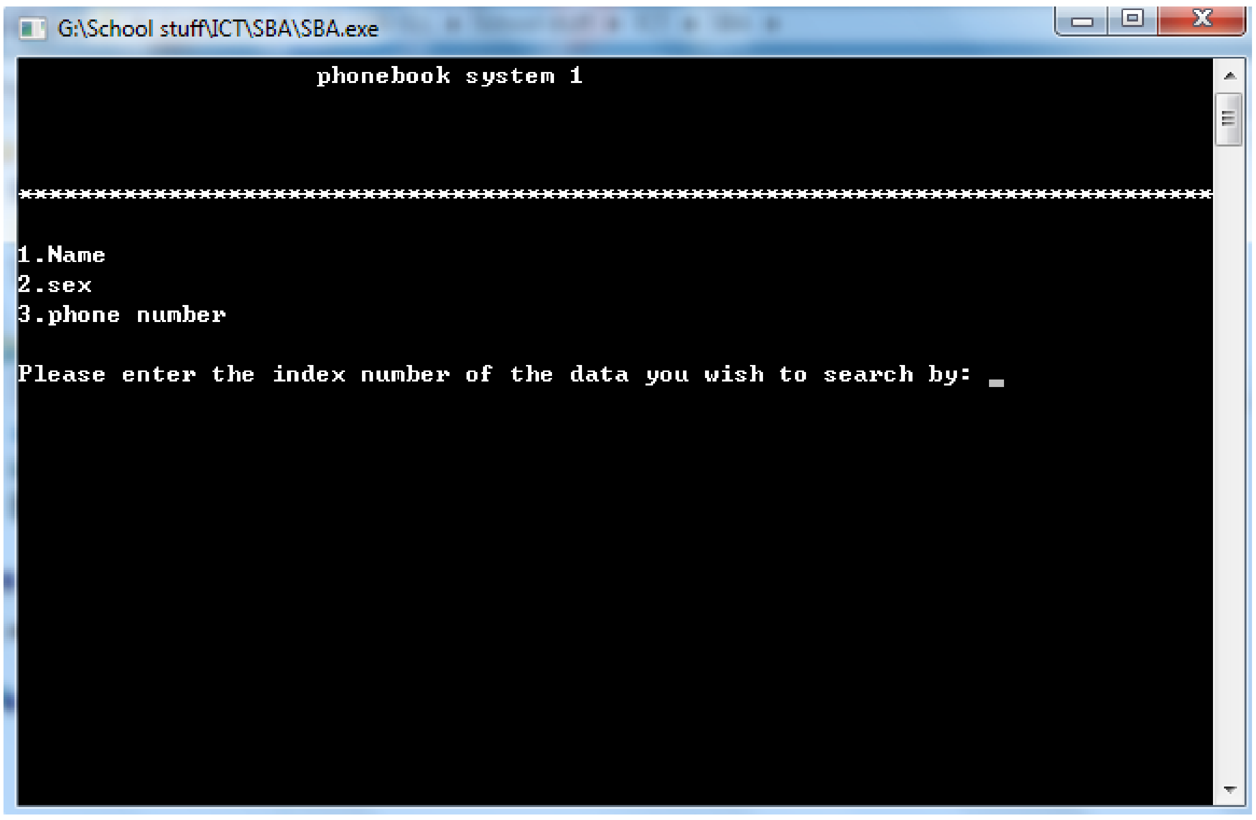
```
                    phonebook system 1



********************************************************************************

                    2.chan siu ming    f    98228912asdfsdf   gmail.com
Enter the contact new name: chan tai man
Enter the contact new gender: f
Enter the contact new phone number: 79879878
Enter the contact new email: dasdasd@dasd.com
record updated
Do you wish to edit again?(Y/N)
n
Press again to return to main menu
```

4)edit record



```
********************************************************************************

contacts
name             sex phonenumber         email
1.   au kui yuen   f    12345678         AKY@gmail.com
2.chan siu ming    f    98228912         asdfsdf   gmail.com
3.Kui              m    12345678         ajsdkljaf@gmail.com
4.kui              m    12434545         dasgagw@yahoo.com
5.kuiyuen          f    99999999           dasjkfha
6.kui              m         999            fasfasf
7.jj               m         999           csfafafg
8.kcuf             F         321                asd
9.gg               m         999              dasda
10.dasd             m         das                 das
11.lai wai          f    98765432            dasdasd
12.dasdasd          f         123             fasfaf
13.chan tai man     f    79879878         dasdasd@dasd.com
please enter the contact name you wish to delete: chan siu ming
2.chan siu ming    f    98228912         asdfsdf   gmail.com
which contact do you wish to delete?: 2
Are you sure to delete this contact?(Y/N):y
contact sucessfully deleted
press again to return to main menu
```

5)delete record



```
                    phonebook system 1



********************************************************************************

1.Name
2.sex
3.phone number

Please enter the index number of the data you wish to search by: _
```

6)search record

Chapter 4 Testing & Evaluation

4.1 Brief Description

In this chapter, I aims to find out any possible bugs, in terms of both logical and run-time errors. In the process I will evaluate the efficiency and ability of the program in achieving its original aims and purposes. The user-friendliness should also be evaluated in the process. After the testing, the program will be debugged and improved according to the result.

4.2 Testing and Evaluation Plan

The program will be tested and evaluated in the following plans:

1) Internal testing and evaluation

This internal test will be performed by the programmer and designer, which is me. A series of testes will be done in order to test the program throughly. This includes testes such as valid input, invalid input, and extreme data input. This is in the aim of testing if the program is able handle all king of data input.

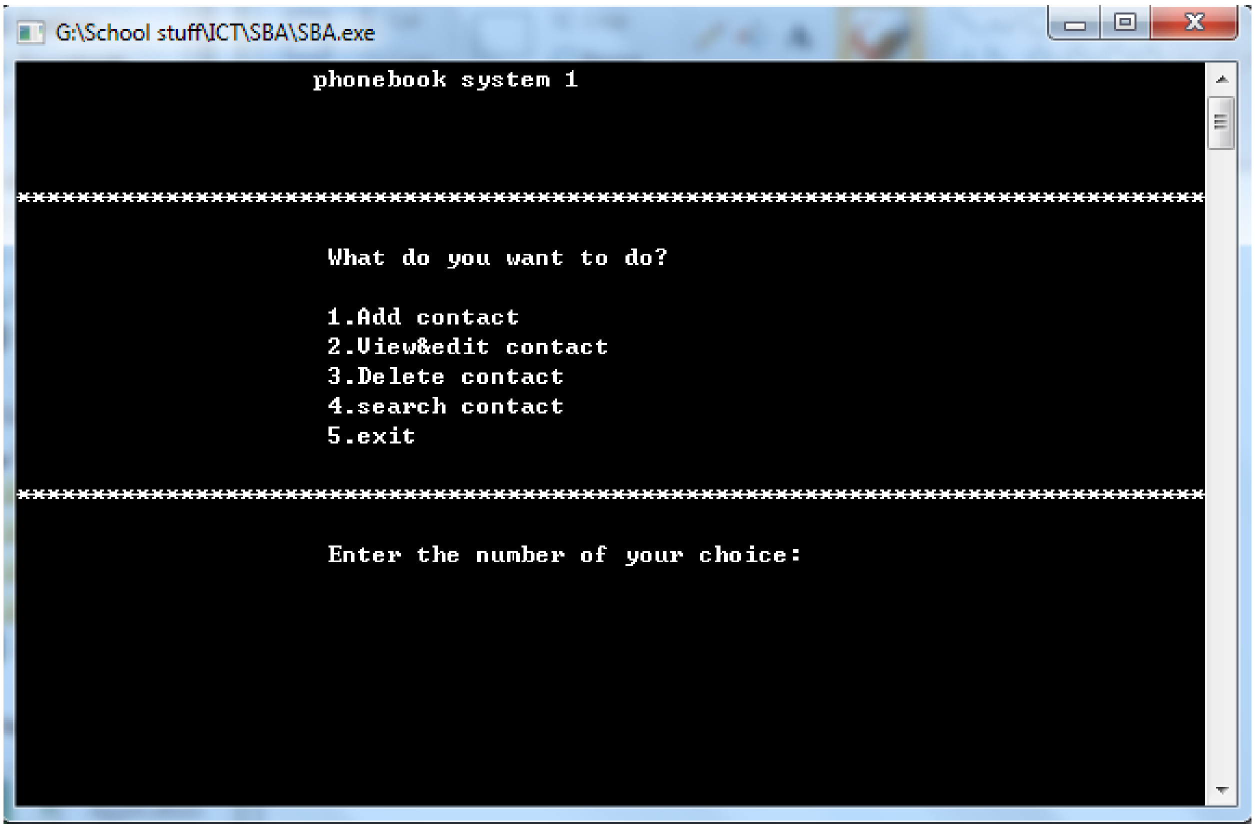
The user friendliness will also be tested by the by the test according to it performance, flexible for further development, edit ability for future program codes etc.

2) External testing and evaluation

after the internal testing, the program will be given by a specific group of user to test the program. The invited user is chosen to be in very similar requirement with Mr.Chan. testers will be invited to fill in a report form according to their experience on the program.

The program will be modified according to the result.

4.3 Internal Testing

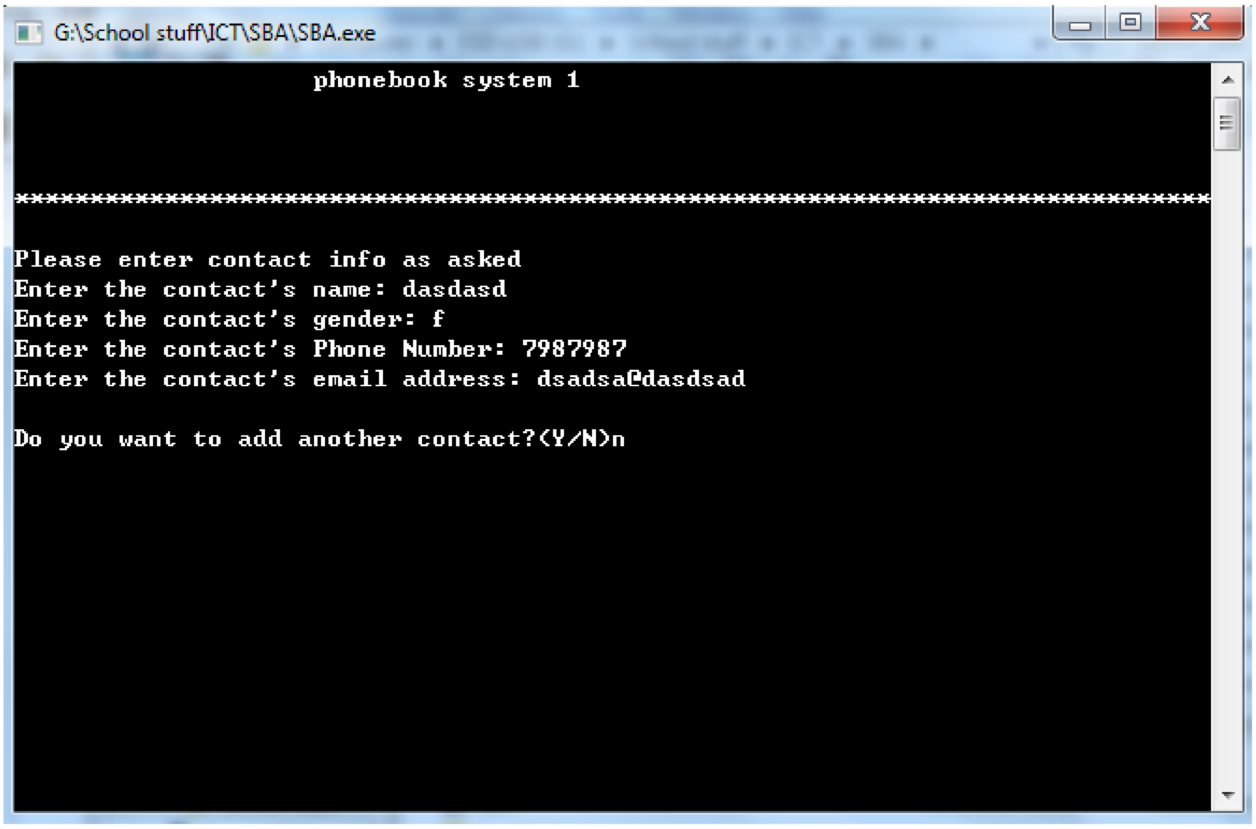


```
                    phonebook system 1


********************************************************************************

                     What do you want to do?

                     1.Add contact
                     2.View&edit contact
                     3.Delete contact
                     4.search contact
                     5.exit

********************************************************************************

                     Enter the number of your choice:
```

1) Test on the main menu

Input	Type of input	Expected result	Actual result	Test result
1	valid input	launch add contact function	match with expected	pass
2	valid input	launch view&edit function	match with expected	pass
3	valid input	launch delete contact function	match with expected	pass
4	valid input	launch search contact function	match with expected	pass
5	valid input	Ask user to press again to leave program	match with expected	pass
Any characters other than 1-5	invalid input	sentence 'invalid input' shown	match with expected	pass
more than one input found	invalid input	sentence 'invalid input' shown	only the first character in the input is read	pass

Main menu test passes.



G:\School stuff\ICT\SBA\SBA.exe

```
                         phonebook system 1



********************************************************************************

Please enter contact info as asked
Enter the contact's name: dasdasd
Enter the contact's gender: f
Enter the contact's Phone Number: 7987987
Enter the contact's email address: dsadsa@dasdsad

Do you want to add another contact?(Y/N)n
```

2) Test on the add contact function

Test on name input column:

Input	Type of input	Expected result	Actual result	Test result
input under 16 character (limit of string)	valid input	allowed, go on to input gender	match with expected	pass
input over 16 character (limit of string)	invalid input	unknown	only the first 16 characters will be recorded	pass

Test on the gender column:

Input	Type of input	Expected result	Actual result	Test result
m,M,f,F	valid input	allowed, go on to input phone number	match with expected	pass
anything other than m,M,f,F	invalid input	‘invalid input, only m,M,f,F is allow’ shown	match with expected	pass

Test on the phone number column:

Input	Type of input	Expected result	Actual result	Test result
integer entered	valid input	allowed, go on to input email	match with expected	pass
Non-integer entered	invalid input	invalid input, only numbers are allowed’ shown	match with expected	pass
over 8 integer entered(limit of string)	invalid input	unknown	only the first 8 integer is recorded	pass

+ The ‘over 8 integer entered’ situation is considered a pass due to the fact that users are not expected to enter any phone number exceeding 8 integer to the phonebook, which is mentioned in the user manual.

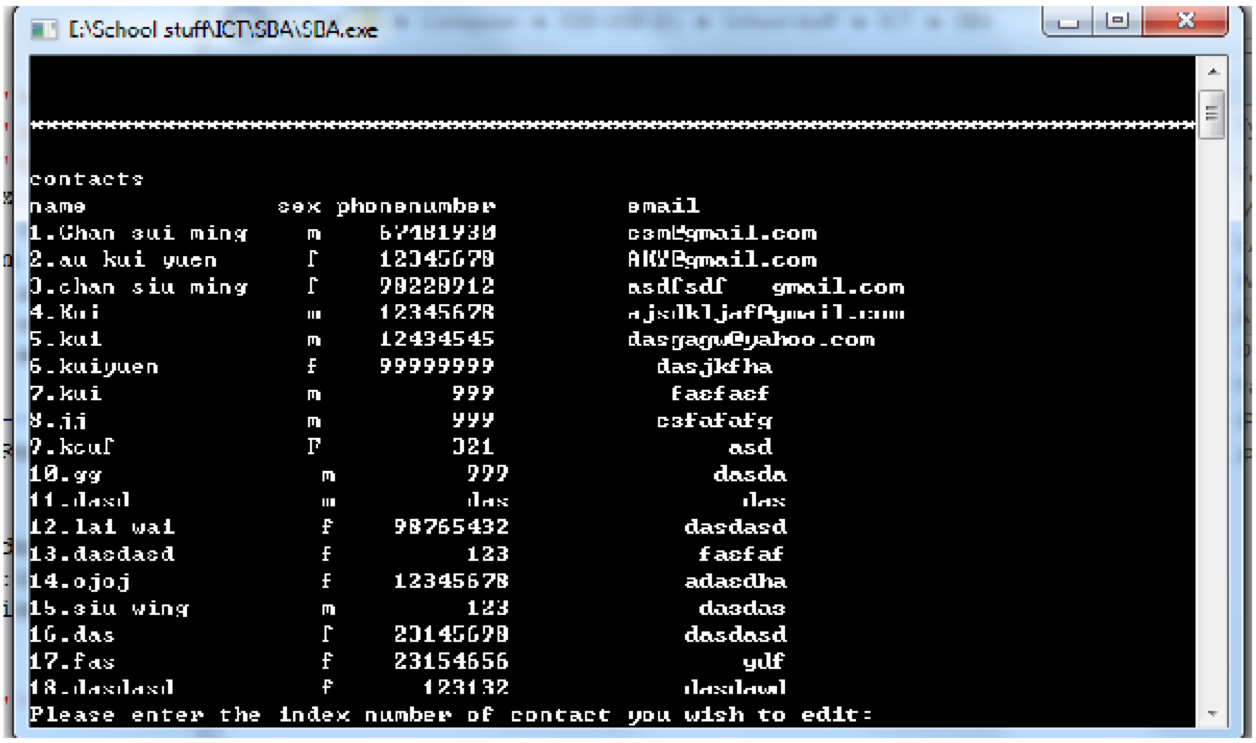
Test on the email column:

Input	Type of input	Expected result	Actual result	Test result
data under 30 characters is entered(limit of string)	valid input	allowed	match with expected	pass
data over 30 characters is entered(limit of string)	invalid input	unknown	allowed, only the first 30 input will be recorded	pass

Test on the ‘Do you want to enter another contact?(Y/N)’ column:

Input	Type of input	Expected result	Actual result	Test result
y,Y	valid input	repeat the add contact function	match with expected	pass
n,N	valid input	back to main menu	match with expected	pass

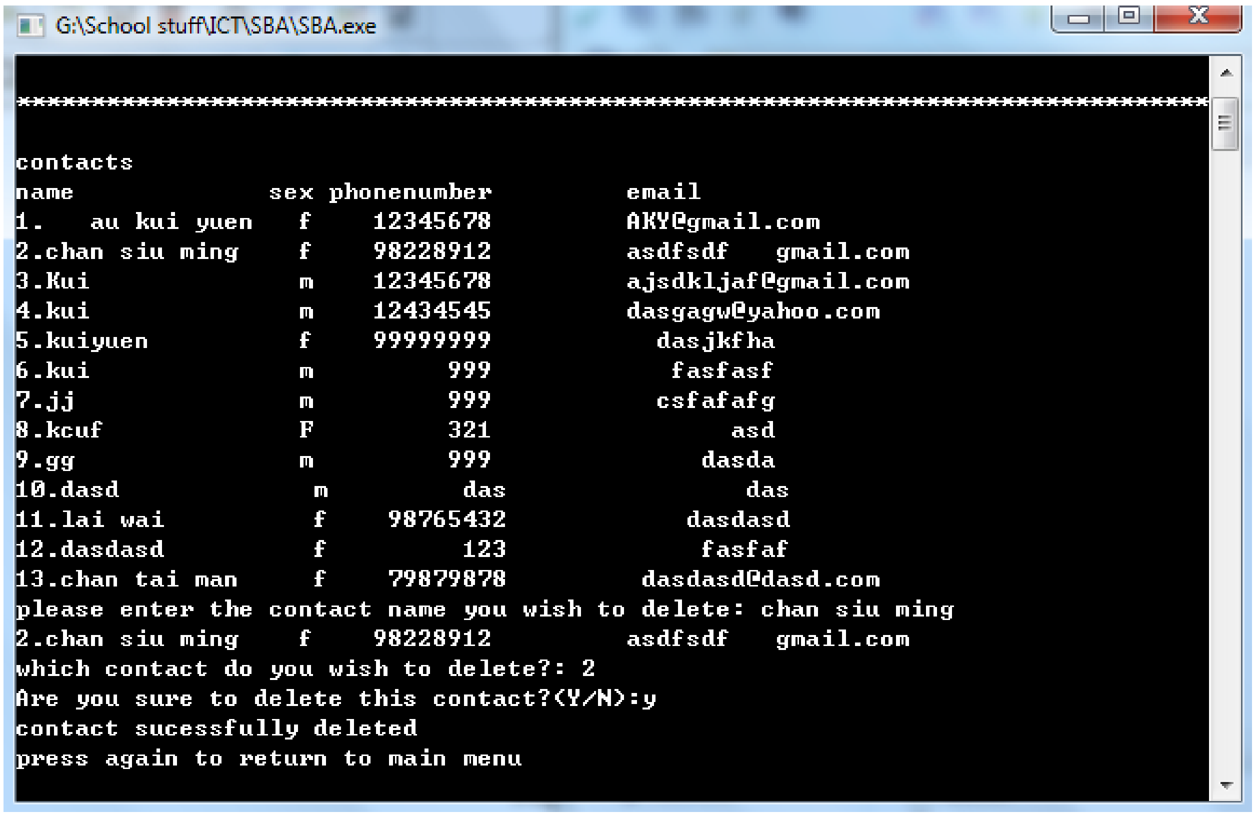
Add contact function test pass.



```
C:\School stuff\ICT\SBA\SBA.exe

********************************************************************************

contacts
name                sex phonenumber          email
1.Chan sui ming      m    67181930          csm@gmail.com
2.au kui yuen        f    12345678          AKY@gmail.com
3.chan siu ming      f    78228712          asdfsdf   gmail.com
4.Kui                m    12345678          ajsdkljaf@gmail.com
5.kui                m    12434545          dasgagw@yahoo.com
6.kuiyuen            f    99999999            dasjkfha
7.kui                m         999             fasfasf
8.jj                 m         999            csfafafg
9.kcuf               F         321                 asd
10.gg                 m         777                dasda
11.dasd               m         das                  das
12.lai wai            f    98765432              dasdasd
13.dasdasd            f         123               fasfaf
14.ojoj               f    12345678              adasdha
15.siu wing           m         123               dasdas
16.das                f    23145678              dasdasd
17.fas                f    23154656                 ydf
18.dasdasd            f      123132              dasdawd
Please enter the index number of contact you wish to edit:
```

3) test on search and edit function

Input	Type of input	Expected result	Actual result	Test result
numbers shown as a index number	valid input	show the found result	match with expected	pass
numbers not shown as a index number	invalid input	contact not found' shown	match with expected	pass

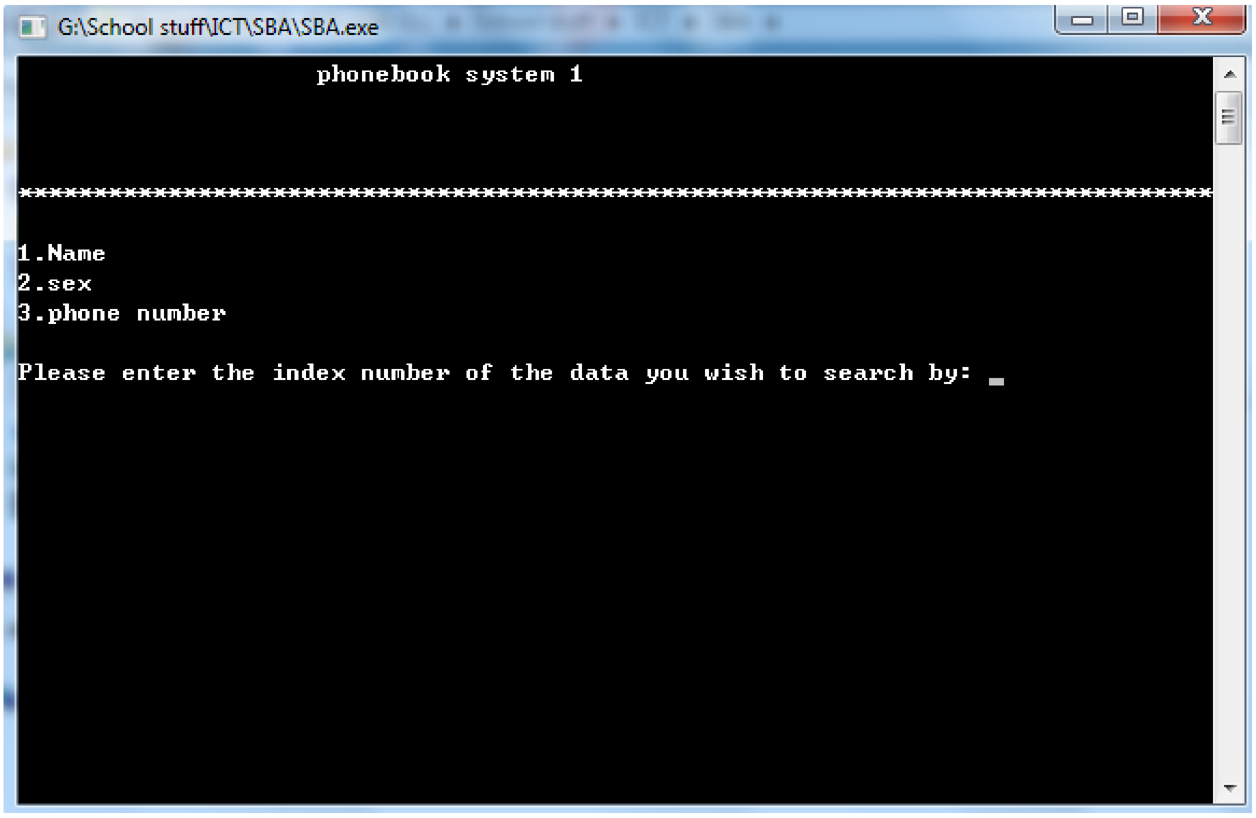


```
G:\School stuff\ICT\SBA\SBA.exe

********************************************************************************

contacts
name                 sex phonenumber         email
1.    au kui yuen     f     12345678         AKY@gmail.com
2.chan siu ming      f     98228912         asdfsdf    gmail.com
3.Kui                m     12345678         ajsdkljaf@gmail.com
4.kui                m     12434545         dasgagw@yahoo.com
5.kuiyuen            f     99999999           dasjkfha
6.kui                m          999            fasfasf
7.jj                 m          999           csfafafg
8.kcuf               F          321                asd
9.gg                 m          999              dasda
10.dasd               m          das                das
11.lai wai            f     98765432            dasdasd
12.dasdasd            f          123              fasfaf
13.chan tai man       f     79879878         dasdasd@dasd.com
please enter the contact name you wish to delete: chan siu ming
2.chan siu ming      f     98228912         asdfsdf    gmail.com
which contact do you wish to delete?: 2
Are you sure to delete this contact?(Y/N):y
contact sucessfully deleted
press again to return to main menu
```

4) test on the delete contact function

test on the input name column

Input	Type of input	Expected result	Actual result	Test result
correct name	valid input	accepted, show next step	match with expected	pass
incorrect name	invalid input	contact not found' shown	match with expected	pass


G:\School stuff\ICT\SBA\SBA.exe

```
                    phonebook system 1



********************************************************************************

1.Name
2.sex
3.phone number

Please enter the index number of the data you wish to search by: _
```

5) test on the search contact function

Input	Type of input	Expected result	Actual result	Test result
1,2,3	valid input	accepted, move on to the next step	match with expected	pass
non-1,2,3 character or integer	invalid input	unknown	ask user if he wish to search again	pass

4.4 Self-Evaluation

After testing the program, which allows me to throughly use and experience the program once, I have found out some positive features and shortcoming in my program.

Firstly, the program provides great directives, which can be easily understood by user, lowering the chances of having the user confused, which is one of the especially important point wile considering the program is designed for a person, Mr.Chan, who is not familiar with technology and computer program at all. The program is very simple, yet direct, providing great efficiency on using the program, eliminating the needs of requiring the user to wonder how to use the program.

Secondly, the stability and performance are great on the program. It provides great respond speed, and no visible and noticeable loading time is required, which once again improve the efficiency of the user using the program. Resources are greatly managed and utilised, while running, the program takes up a tiny amount of RAM, the size of the program is also great, requiring only 36KB for the program and 88KB for the entire file containing the source code and external data file.

Thirdly, the system provides great flexibility and expandability for programmers on future development for the program. New functions can easily be added to the main body of the program, due to the usage of **case** function on selecting the functions.

On the other side of the hill, there are indeed some shortcoming in the program. It starts of from the 'not really attractive' interface, while the main menu and user interface provides great directions and instruction to the user, the user interface isn't particularly good looking, due to the limitation of my programming skills. Also, the window size can't be adjusted by the user, due to the limitation of window exe.

Last but not least, the program lacks a function for the user to exit to the main menu as desired, due to the requirement of huge amount of tuning. In order for the user to go back to the main menu, they will have to finish all the process in the selected function, which sometimes can be annoying

4.5 External Testing and Evaluation

External testing will be done by a group of people, which has similar situation with Mr.Chan, mid-aged, not familiar with computer program and using the program for both work and personal. I will provide the program and a evaluation questionnaire to the testers, and allow them to use the program for a period of time, ranging from 2 to 10 days. A sample of the evaluation questionnaire is provided in the appendices.

Here are the summary report of the collected data from 10 participant. The average score is corrected to the nearest integer.

	average score(10)
The program is user-friendly	7
The program is provides great direction, not confusing	7
The program is easy to use	8
The usefulness of the functions	7
The performance is good	8
The stability is good	6

Some testers does report a lack of security in the text file, which is true.

Chapter 5 Conclusion & Discussion

5.1 Pros and cons of my Program

I am nowhere near a experienced programmer, and in the situation where professional programmer have pros and cons in their program, it is inevitable for my program to be not so perfect, but everything has its ups and downs, not a exception for my program. Here are the pros and cons of my program.

Lets start with the good things, this program provide great user-friendliness, reflected by the external testers, the program is very easy to use, even for a first timer, by providing great and useful direction to the user, as an example, the main menu requires the user to just enter the index number of the desired functions, which is extremely easy to understand, actually even my 8 year old cousin is able to use the program, which proves the user-friendliness and easiness in control of the program. This concludes in my opinion the first pros of my phonebook program.

Secondly, the variety of functions provided in the program is very sufficient, the functions basically covers all the basic needs of the potential user, who is people with similar situation with Mr.Chan. there are also interaction between different function, which provide some great convenience for the user, for example, in the search function, if the user searches a contact which is non-existing in the database, the program will immediately ask if the user wish to add it at once, if the answer is yes, then he will be transferred into the add contact function. In this way the user saves the time required for moving between functions.

Thirdly, the program provides a great flexibility and expandability for any future development, either by me, the original programmer, or even other programmers. The first reason being that the program’s source code is tidily and written in a very organised way, missions are clearly distributed by different procedures, with that, new functions and procedure can easily be added into the program. Also, description are provided in some unique programming style by my, which eliminates most of the possible confuses of future programmer.

Everything has its pros and cons, my program is no exception. Quite frankly, the program's function is still only simple, as mentioned above, indeed it can fulfil the basic needs of the user, but it ends here, many functions is still not available for the user, as an example, the manual save function is not available, the program will only save every time the user leave the program, which will be a huge inconvenience for the user.

Also, the data slot available for the user to enter for the contact's data is limited to the provided four, which includes name, gender, phone number and email. The program won't be able to record anything about the contact other than those four. It could be a huge inconvenient for user.

Not to mention the bore of the program user-interface, it is quite boring honestly, indeed it is able to achieve what it needs to, but it just looks kind of dull in my opinion.

Last but not least, the contacts can only be stored in a pure verbal way, no multimedia element can be stored in the data of the contact, which might cause some tiny inconvenience for the user, although the problem is not something huge, it might still create inconvenience to user, which i consider as a problem.

Quite honestly this program is nothing near perfect, but it is still made by my full effort, and I hope for better next time.

5.2 Future Improvement

After handing the program to Mr.Chan, I will continue to update the program with variety of features, according to either the response of Me.Chan, and my own ideas. If any bug shows up, I will make sure to release patches as soon as possible. In order to provide the best and most enjoyable experience with the program.

Regarding the above mentioned shortcoming of the program, i will continue to try my very best to provide the best solution to them. In the future, I aims to add a manual save and update feature to the program, which enhances the usability of the program, and also improve the convenience of the program provided to the user. After that, I hopes to provide the program with a update which provide more data to be entered to the contact's data, currently I am considering the possibility of allowing user to create own data type for all contacts, according to their own needs, for example, Mr.Chan can create a new column called 'work phone number', which allows him to record the work phone number of contact, and use the original type for personal phone number. At the mean time I am not sure either pascal language allows it, and if my programming skills will be able to achieve that, but I will still try my best to implement this feature, in order to provide a better experience to users.

Regarding the user-interface problem, quite frankly it is a innate limitation to the pascal program language, which is not being able to store any form of media file. In the future I will continue to release patches to enhance the user interface, with what I am able to do with the pascal language. I might even transfer the program into a more advanced programming language, but that would be a very faraway plan, and it is currently still in idea stage.

5.3 Self-Reflection

The program has been worked on for about half a year, which is indeed a lot of work and effort, from planning, designing, coding to testing, making report, it has been quite a journey. In the pass, I have never personally developed a full-fledged program, and quite frankly, I am not a experienced programmer at all, which leaves the program to be in a not-so-perfect form. But indeed, I have learned a whole lot from developing this program.

Before developing this program, to be completely honest i don't even know the existence of procedure, this program has attracted me to investigate into programming, and has raised my interest of programming a lot, in the past I thought programming is some extremely boring and hard and nerdy thing, but after developing this program, i found out that it is to some extent very challenging, with countless time of satisfaction, frustration, anticipation, I bet that me in the past cant imagine that I am actually quite enjoying programming.

Also i have learnt the potential of a single human being facing deadlines, the ability to face deadlines is extremely impressive. Of course being a deadline fighter isn’t exactly a good thing, next time i will manage my time distribution better.

Chapter 6 Reference and Acknowledgement

From Internet websites

1. http://www.freepascal.org/

2. https://hk.answers.yahoo.com/

From books

1. New Senior Secondary Information and Communication Technology elective D1

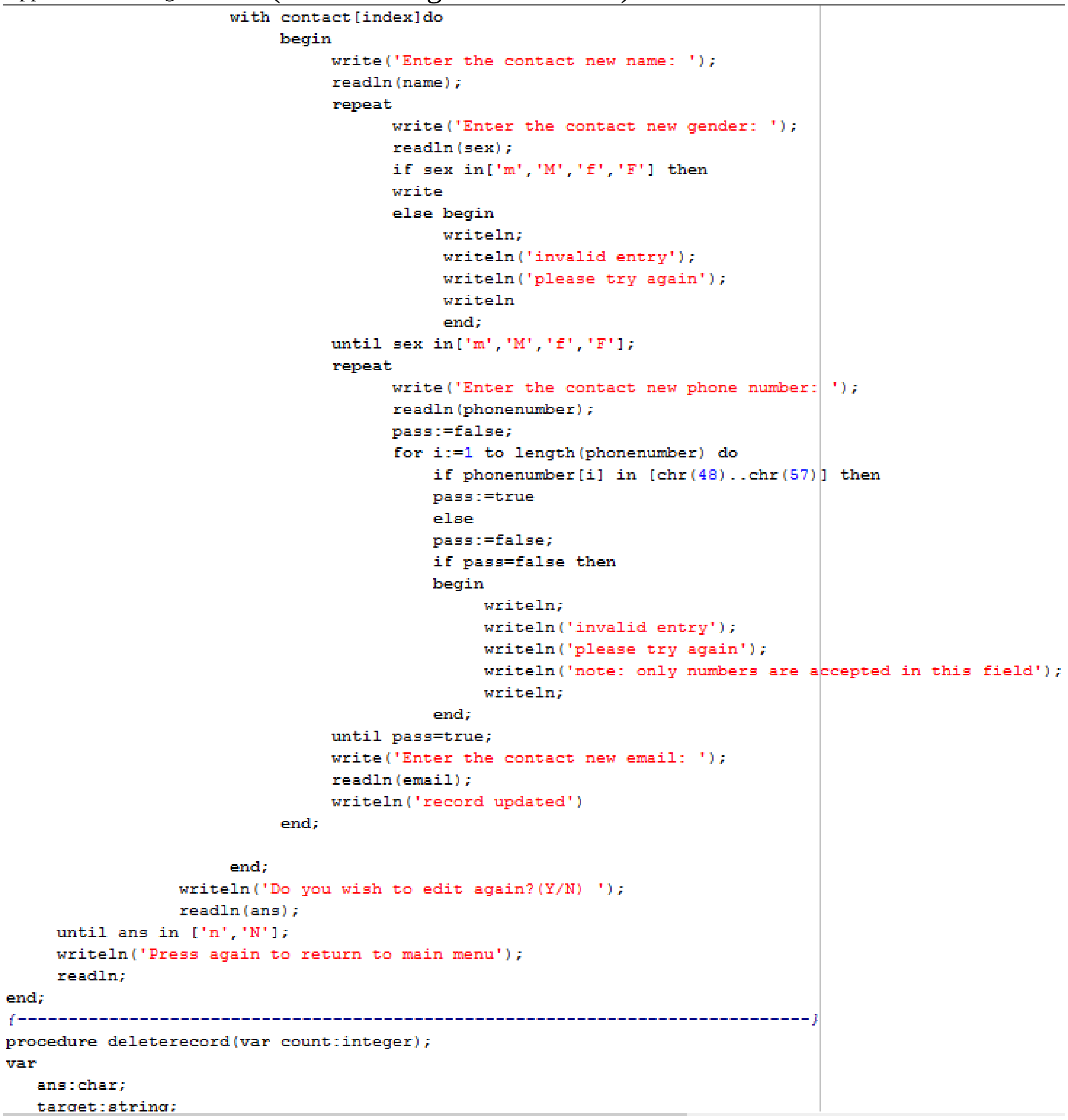
Acknowledgement

Here I would like to express a huge thanks for Mr.Chu, my ICT teacher for spending countless amount of time help with this project, this program won’t be a thing without his help. Also his generous extent of the program deadline saved me a lot of time.

Also the testers of the program is also greatly appreciated for their generous help.

Appendices

Appendix 1 – Program Code (after Testing & Evaluation)



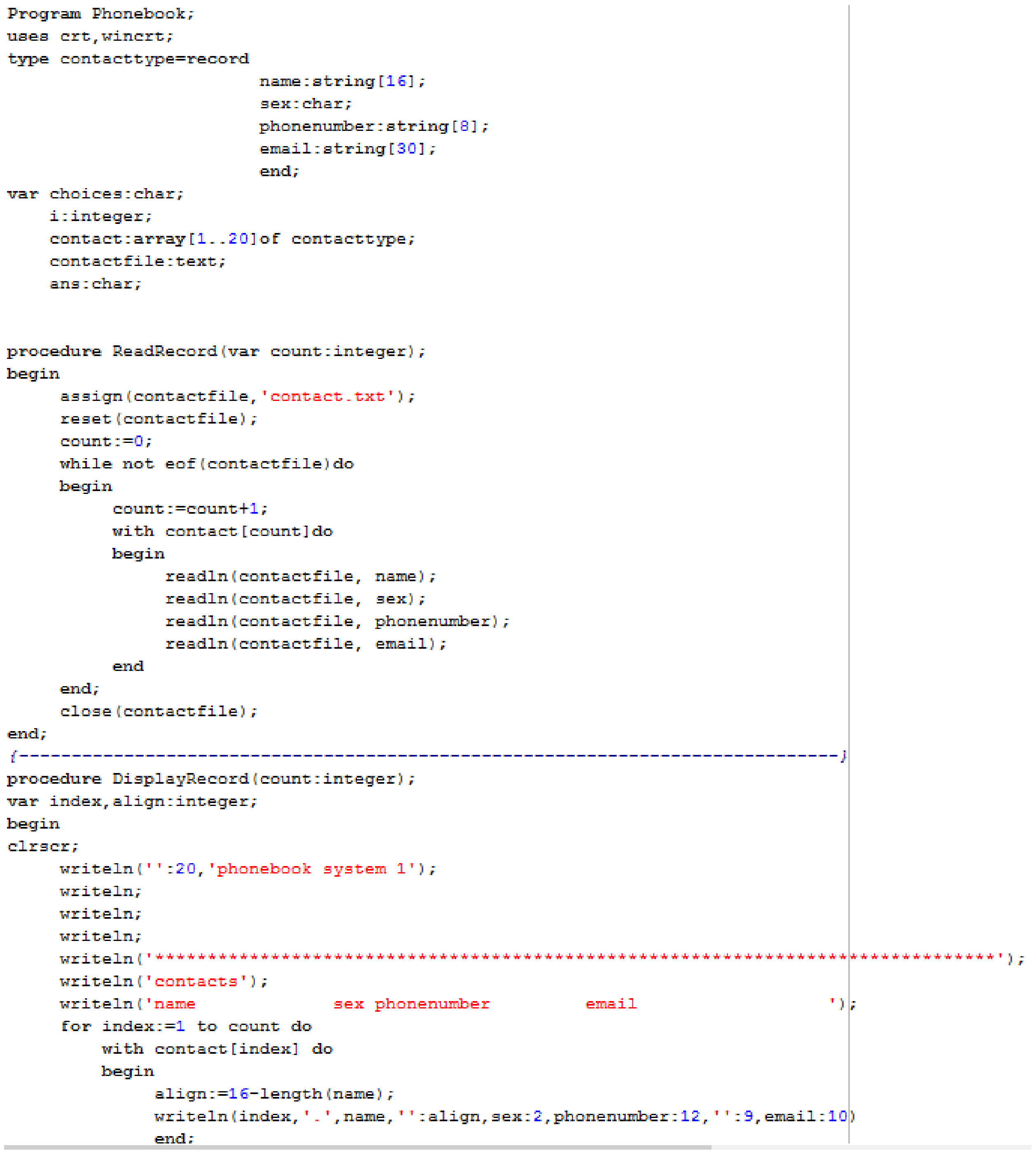
```
                      with contact[index]do
                           begin
                                write('Enter the contact new name: ');
                                readln(name);
                                repeat
                                      write('Enter the contact new gender: ');
                                      readln(sex);
                                      if sex in['m','M','f','F'] then
                                      write
                                      else begin
                                           writeln;
                                           writeln('invalid entry');
                                           writeln('please try again');
                                           writeln
                                           end;
                                until sex in['m','M','f','F'];
                                repeat
                                      write('Enter the contact new phone number: ');
                                      readln(phonenumber);
                                      pass:=false;
                                      for i:=1 to length(phonenumber) do
                                          if phonenumber[i] in [chr(48)..chr(57)] then
                                          pass:=true
                                          else
                                          pass:=false;
                                          if pass=false then
                                          begin
                                               writeln;
                                               writeln('invalid entry');
                                               writeln('please try again');
                                               writeln('note: only numbers are accepted in this field');
                                               writeln;
                                          end;
                                until pass=true;
                                write('Enter the contact new email: ');
                                readln(email);
                                writeln('record updated')
                           end;

                      end;
                 writeln('Do you wish to edit again?(Y/N) ');
                 readln(ans);
     until ans in ['n','N'];
     writeln('Press again to return to main menu');
     readln;
end;
{-------------------------------------------------------------------------------}
procedure deleterecord(var count:integer);
var
   ans:char;
   target:string;
   found:boolean;
   i,num,index,align:integer;
begin
     write('please enter the contact name you wish to delete: ');
     readln(target);
     found:=false;
     for i:=1 to count do
         with contact[i] do
              if upcase(name)=upcase(target)
                 then begin
                      found:=true;
                      with contact[i] do
                      begin
                           align:=16-length(name);
                           writeln(i,'.',name,'':align,sex:2,phonenumber:12,'':9,email:10);
                      end;
            if not found
            then begin
                 writeln('contact not found');
                 readln
                 end
            else begin
            with contact[i] do begin
            {align:=16-length(name);
            writeln(i,'.',name,'':align,sex:2,phonenumber:12,'':9,email:10);   }
            found:=true
            end;
            end;
            write('which contact do you wish to delete?: ');
            readln(i);
            with contact[i]do
                 write('Are you sure to delete this contact?(Y/N):');
                 readln(ans);
                 if ans in ['Y','y']
                    then begin
                         with contact[i] do
                              name:='deleted contact';
                         //     count:=count-1;
                              writeln('contact sucessfully deleted');
                              writeln('press again to return to main menu');
                              readln
                    end
                 else begin
                 writeln('press again to return to main menu');
                 readln
                 end
      end
end;
{---------------------------------------------------------------------------------}
procedure writerecord(count:integer);
var n:integer;
```

```
begin
     rewrite(contactfile);
     for n:=1 to count do
         with contact[n] do
         if name<>'deleted contact'
            then begin
                      writeln(contactfile, name);
                      writeln(contactfile, sex);
                      writeln(contactfile, phonenumber);
                      writeln(contactfile, email);
                 end;
close(contactfile);
         end;
{------------------------------------------------------------------------------}
procedure searchrecord(var count:integer);
var
   target:string;
   found:boolean;
   i,index,align:integer;
   ans:char;
begin
     repeat
     clrscr;
     writeln('':20,'phonebook system 1');
     writeln;
     writeln;
     writeln;
     writeln('********************************************************************************');
     writeln('1.Name');
     writeln('2.sex');
     writeln('3.phone number');
     writeln;
     write('Please enter the index number of the data you wish to search by: ');
     readln(index);
     case index of
          1: begin
                  write('please enter the name of contact you wish to view: ');
                  readln(target);
                  found:=false;
                  for i:=1 to count do
                      begin
                      with contact[i] do
                           if  pos(upcase(target),upcase(name))=1 then
                           begin
                                found:=true;
                                align:=16-length(name);
                                writeln(i,'.',name,'':align,sex:2,phonenumber:12,email:10);
                           end
                      end;
                  if not found then
                     begin
```

```
                    writeln('contact not found');
                    readln;
                    writeln('Do you wish to add this record? ');
                    readln(ans);
                    end;
                 if ans in ['y','Y'] then
                       addrecord(i)
                    else readln;
                 if found then
                 readln;
                 end;
          2:begin
                     write('please enter the sex of contact you wish to view: ');
                     readln(ans);
                     found:=false;
                     for i:=1 to count do
                     begin
                     with contact[i] do
                          if  pos(upcase(ans),upcase(sex))=1 then
                          begin
                               found:=true;
                               align:=16-length(name);
                               writeln(i,'.',name,'':align,sex:2,phonenumber:12,email:10);
                          end
                     end;
                     if not found then
                        begin
                        writeln('contact not found');
                        readln;
                        writeln('Do you wish to add this record? ');
                        readln(ans);
                     if ans in ['y','Y'] then
                        addrecord(i)
                        else begin
                        writeln('press again to return to main menu');
                        readln
                        end;
                     end;
                     if found then
                     readln;
                     end;
                3:begin
                         write('please enter the phone number of contact you wish to view: ');
                         readln(target);
                         found:=false;
                         for i:=1 to count do
                         begin
                         with contact[i] do
                              if  pos(upcase(target),upcase(phonenumber))=1 then
                              begin
                               found:=true;
```

```
                                  align:=16-length(name);
                                  writeln(i,'.',name,'':align,sex:2,phonenumber:12,email:10);
                                 end
                            end;
                            if not found then
                            begin
                                 writeln('contact not found');
                                 readln;
                                 writeln('Do you wish to add this record? ');
                                 readln(ans);
                                 if ans in ['y','Y'] then
                                    addrecord(i)
                            else begin
                                 writeln('press again to return to main menu');
                                 readln
                                 end;
                            end;
                            if found then
                            readln;
                            end;
                            end;
                            writeln('Do you want to search again?(Y/N): ');
                            readln(ans);
                            until ans in ['n','N'];
                            end;
{-------------------------------------------------------------------------------}
```

```
{------------------------------------------------------------------------------}
begin         {main menu}
     ReadRecord(i);
     repeat
     clrscr;
     textcolor(2);
     writeln('':20,'phonebook system 1');
     writeln;
     writeln;
     writeln;
     writeln('********************************************************************************');
     writeln('':21,'What do you want to do?');
     writeln;
     writeln('':21,'1.Add contact');
     writeln('':21,'2.View&edit contact');
     writeln('':21,'3.Delete contact');
     writeln('':21,'4.search contact');
     writeln('':21,'5.exit');
     writeln;
     writeln('********************************************************************************');
     writeln('':21,'Enter the number of your choice: ');
     write('':21);
     readln(choices);
     case choices of
          '1':begin
                   AddRecord(i);
              end;
          '2':begin
                   searchandEditRecord(i);
              end;
          '3':begin
                   DisplayRecord(i);
                   deleterecord(i);
                   writerecord(i);
                   ReadRecord(i);
                   readln
               end;
          '4':searchrecord(i);
          '5':writeln('':20,'press again to exit the program')
          else begin
               writeln('':21,'invalid option');
               readln
               end
     end;
     until choices='5';
     readln;
end.
```



```
Program Phonebook;
uses crt,wincrt;
type contacttype=record
                        name:string[16];
                        sex:char;
                        phonenumber:string[8];
                        email:string[30];
                        end;
var choices:char;
    i:integer;
    contact:array[1..20]of contacttype;
    contactfile:text;
    ans:char;


procedure ReadRecord(var count:integer);
begin
     assign(contactfile,'contact.txt');
     reset(contactfile);
     count:=0;
     while not eof(contactfile)do
     begin
          count:=count+1;
          with contact[count]do
          begin
               readln(contactfile, name);
               readln(contactfile, sex);
               readln(contactfile, phonenumber);
               readln(contactfile, email);
          end
     end;
     close(contactfile);
end;
{--------------------------------------------------------------------------------}
procedure DisplayRecord(count:integer);
var index,align:integer;
begin
clrscr;
     writeln('':20,'phonebook system 1');
     writeln;
     writeln;
     writeln;
     writeln('********************************************************************************');
     writeln('contacts');
     writeln('name             sex phonenumber         email                  ');
     for index:=1 to count do
         with contact[index] do
         begin
              align:=16-length(name);
              writeln(index,'.',name,'':align,sex:2,phonenumber:12,'':9,email:10)
              end;
         end;
{------------------------------------------------------------------------------}
procedure AddRecord(var count:integer);
var
   ans:char;
   i,align,index:integer;
   found,pass:boolean;
   target:string;
begin
     clrscr;
     writeln('':20,'phonebook system 1');
     writeln;
     writeln;
     writeln;
     writeln('********************************************************************************'
     writeln('Please enter contact info as asked');
     append(contactfile);
     repeat
           count:=count+1;
           with contact[count]do
           begin
                write('Enter the contact''s name: ');
                readln(name);
                repeat
                begin
                      write('Enter the contact''s gender: ');
                      readln(sex);
                      if sex in['m','M','f','F'] then
                      write
                      else begin
                      writeln;
                      writeln('invalid entry');
                      writeln('please try again');
                      writeln
                      end;
                      end
                until sex in['m','M','f','F'];
                repeat
                begin
                     write('Enter the contact''s Phone Number: ');
                     readln(phonenumber);
                     pass:=false;
                     for i:=1 to length(phonenumber) do
                     if phonenumber[i] in [chr(48)..chr(57)] then
                     pass:=true
                     else
                     pass:=false;
                     if pass=false then
                     begin
                     writeln;
                     writeln('please try again');
                     writeln('note: only numbers are accepted in this field');
                     writeln;
                     end;
                     end
                until pass=true;
                write('Enter the contact''s email address: ');
                readln(email);
                writeln(contactfile, name);
                writeln(contactfile, sex);
                writeln(contactfile, phonenumber);
                writeln(contactfile, email);
           end;
           writeln;
           write('Do you want to add another contact?(Y/N)');
           readln(ans)
     until ans in ['N','n'];
     close(contactfile)
end;
{-------------------------------------------------------------------------------}
procedure searchandEditRecord(var count:integer);
var
   target:integer;
   found,pass:boolean;
   i,a,index,align:integer;
   ans:char;
begin

     repeat
           append(contactfile);
           displayrecord(count);
           write('Please enter the index number of contact you wish to edit: ');
           readln(target);
           found:=false;
           for i:=1 to count do
               with contact[i] do
                 if  target=i then
                 begin
                      clrscr;
                      writeln('':20,'phonebook system 1');
                      writeln;
                      writeln;
                      writeln;
                      writeln('********************************************************************************');
                      align:=16-length(name);
                      writeln('':20,i,'.',name,'':align,sex:2,phonenumber:12,email:10);
                      found:=true;
                 end;
                 if not found
                 then writeln('contact not found')
                 else begin
```

Appendix 2 - Working schedule

Date	**Event**
June to Septemb er	planning
October to Novemb er	coding
Novemb er to january	report