

Cheung Sha Wan Catholic Secondary School  
Information & Communication Technology  
School Based Assessment

Title: Phone Book System

# Content

1.Objective and Analysis

2.Design and implementation

3.Testing and Evaluation

4.Conclusion and Discussion

5.Project Management

6.Appendix

7.Source Code

# Objective and Analysis

## 1.1 Background and situation

Nowadays, smart phone become one of the most important things towards people. Less people use phone call to connect people since the popular of online communication software. Many people think that phone call is not necessary anymore because people can chat online and send message quickly through communication software. Although different communication software appeared in mobile apps to the public, phone numbers still a necessary data. As the phone numbers are mostly stored in the SIM card or inside the mobile phone, people may find difficult to maintain the data. As a result, using the program to search and maintain the phone book system can be more efficient and correct the data more details.

## 1.2 Target users & User requirements

The target users of this phone book system are the computer user. It is because the phone book system is only support the computer. The phone book system is decided for the user with basic computer skill. Also, as the phone book system all written in English, users may require with basic English reading skills and understand some basic English vocab.

## 1.3 Considerations

In this program, there are some considerations inside the phone book system. As the phone book system need to make some decision making, it will become one of the considerations. Also, the user-interface may be another consideration because the user-interface is important to the user-friendliness of the program. If the design of the program and the expansion are not well, the useful of the program will be affected. As a result, those considerations will become most important things of my program.

#### 1.4 Software Analysis

In this program, I use the Pascal software to write this phone book system. It is because Pascal software is a simple program language. Also, Pascal is similar with the English writing. As a result, using Pascal to write program can be easier to complete the task. In the Pascal software, I choose the dev-Pascal to write the program coding because the dev-Pascal is a free-software and the user-interface of dev-Pascal is more user-friendly. So, I decided to use the dev-Pascal to do the project.

# Design and implementation

## 1. The User Interface

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

## 2. Modularization

In this program, it has divided into three sub-programs. They are edit, create and delete. In this program, the sub-programs are operating under the searching, which is the main program of this phone book system because the edit and delete function can be call after the searching has been done. So that, the searching function act an important character function in this program. As a result, the searching procedure will be reused in this program for several times.

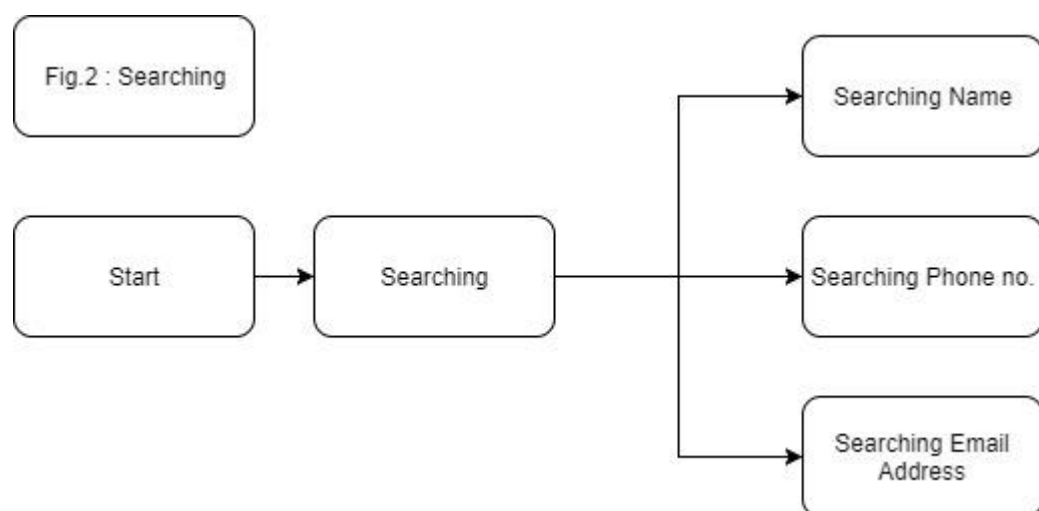
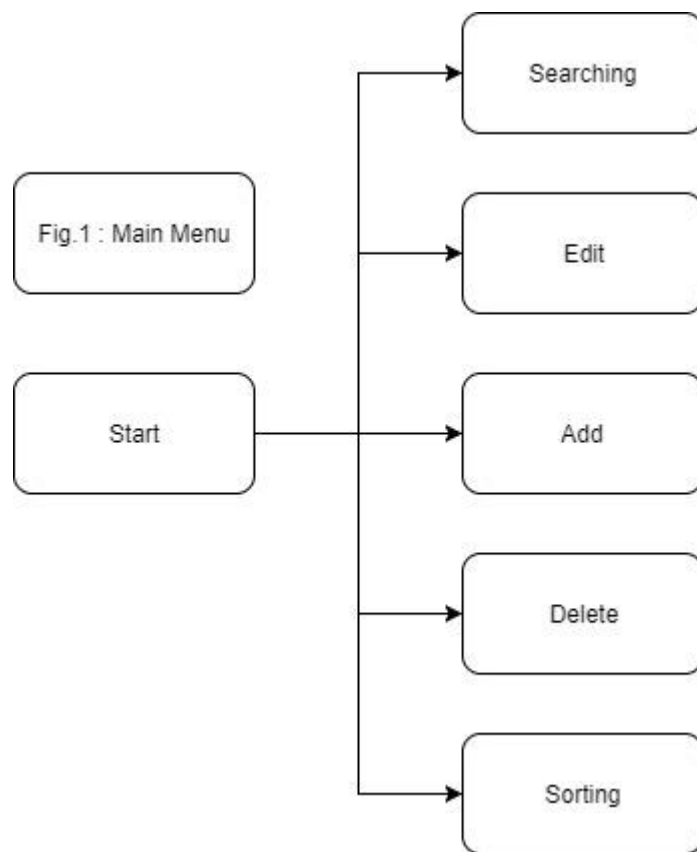


Fig.3 Searching  
Name

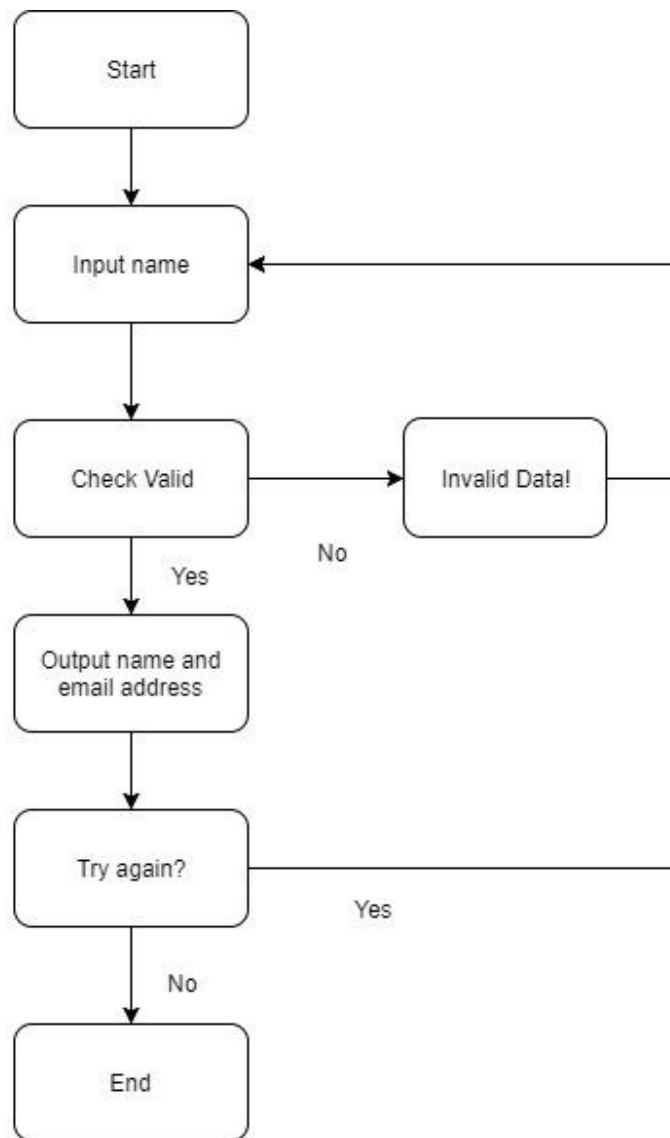


Fig.5 Searching email

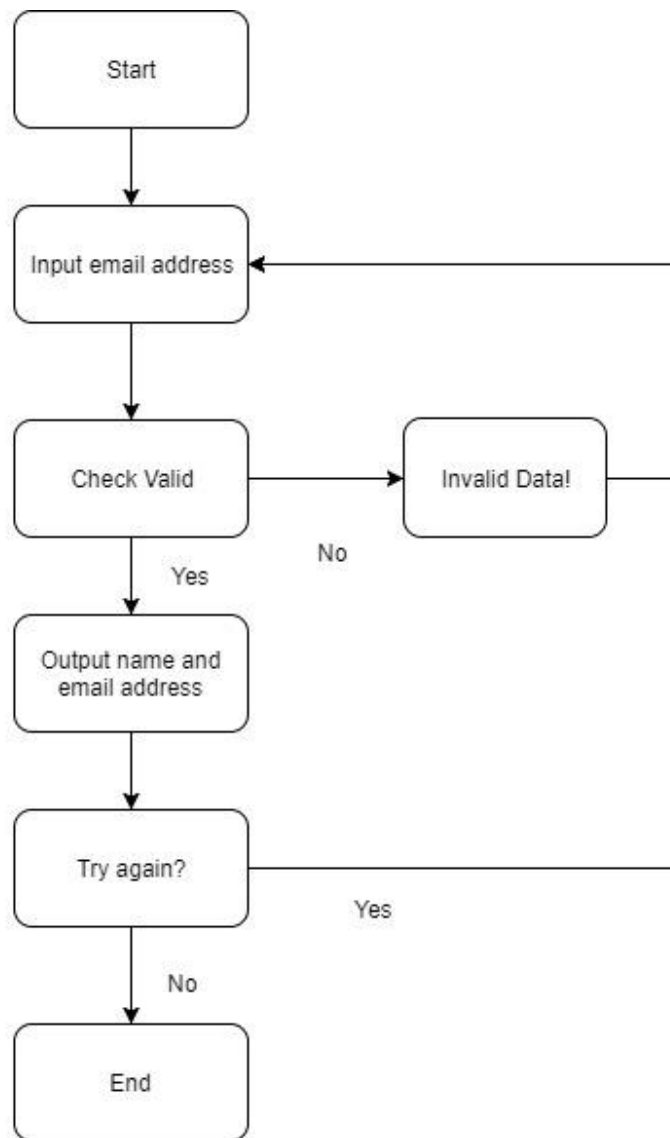




Fig.6 Edit phone book

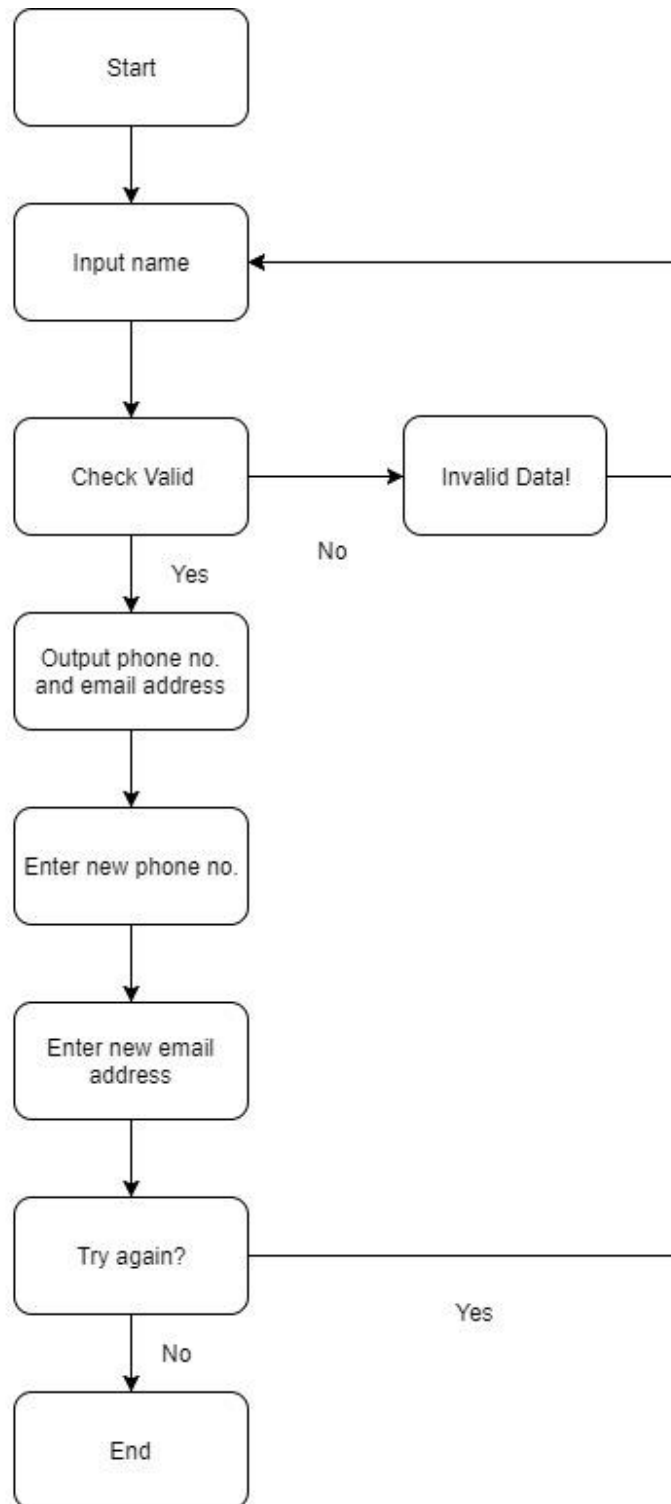


Fig.7 Add new data

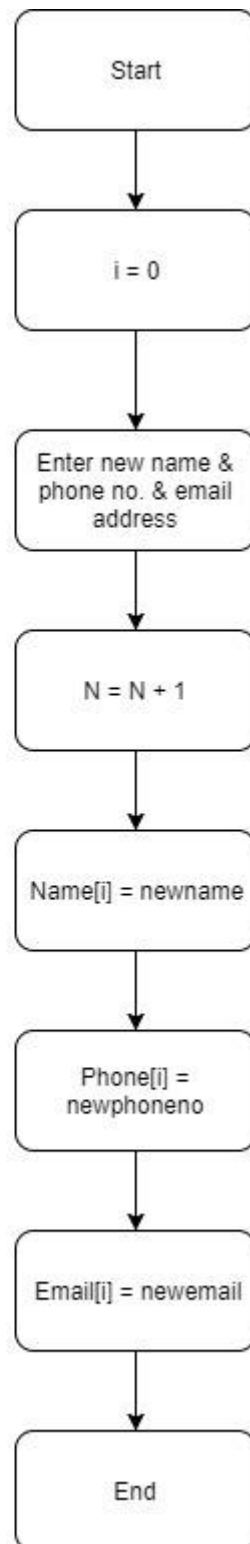


Fig.8 Delete data

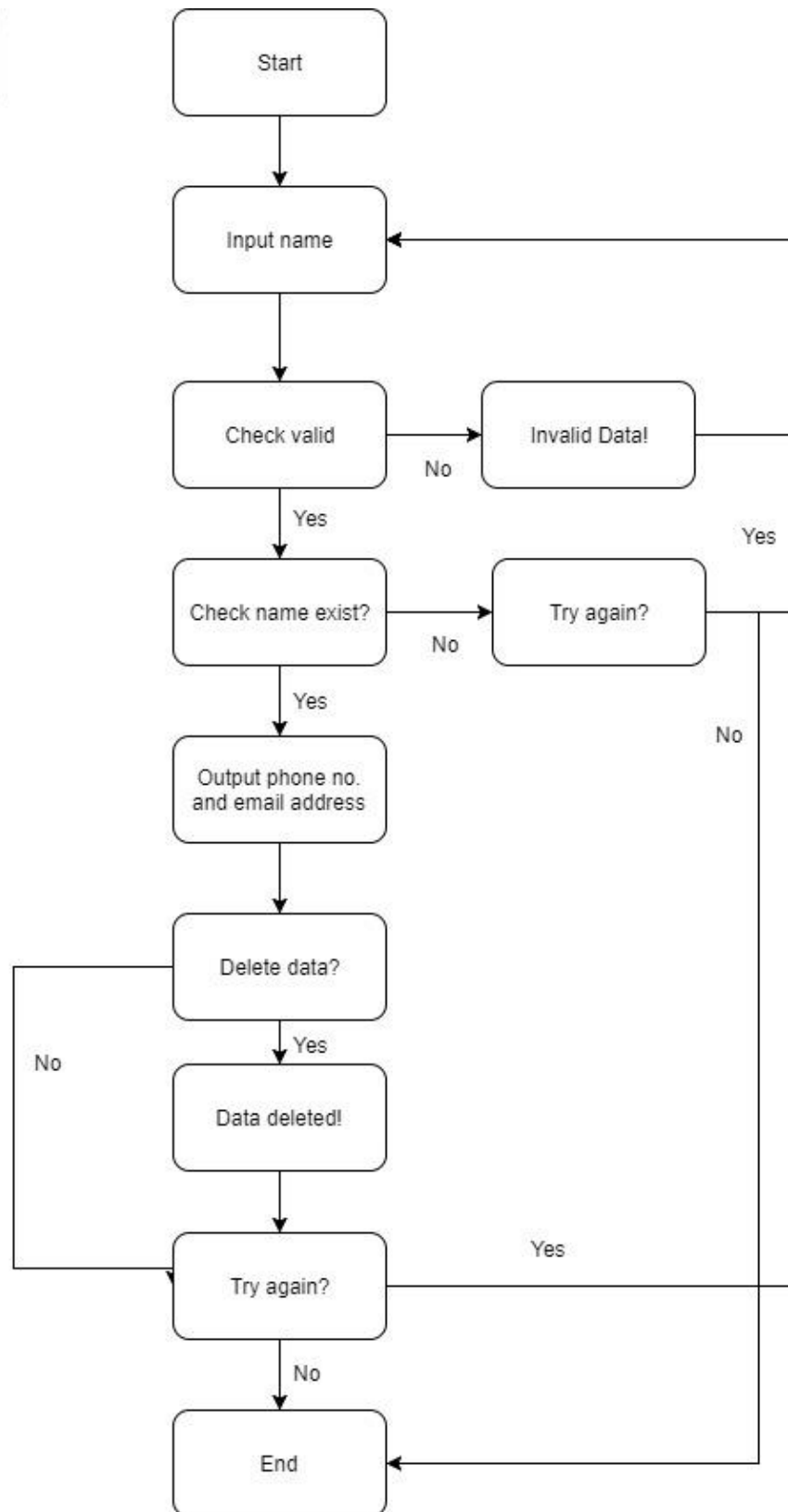
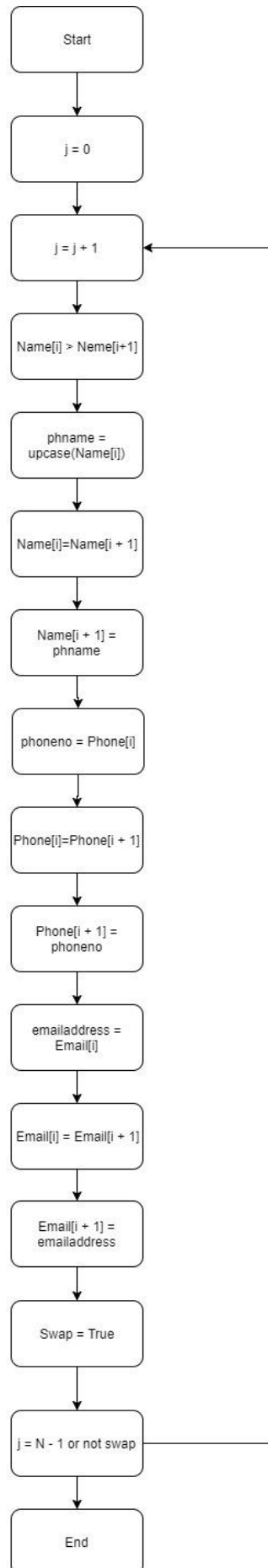


Fig.9 Sorting



### 3. The Data Structure

In this program, I use the file to store the data of the phone numbers, names and email address. Also, in the program, I used three arrays to store the above three data because I think using the file to store the data can be easily to maintain. As the data in the file can be edit and delete easily. So, I tend to use the file to store the data instead of the record. It is using the file can let the program become simple and more easily to debugging and data validation. So that, I prefer using the file to the record. Also, using the array to store the data is more efficient. It is because there are three main data information. As to reduce the redundant of listing the data record, using the array can prevent this problem because when the array is used, the target data can be found by changing the indicator inside the specific array. Therefore, the program can access the data faster and directly. The searching procedure can be operated more efficient due to the redundant program coding is not exist. As a result, I tend to use the array to store the data inside the program.

### 4. Constants and Limitations

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

## 5. Data Control

In order to find the suitable data, there is a data check after the input of the data in the sub-program because the typing error may occur when the user is inputting a large variety of data at the same time. As a result, the data check can help to minimize the problem.

In the data check, there are some requirements for checking the input of the data in each sub-program. First, if user only enter the empty record in the program, the program will show an empty result until user input data inside. When the input is passed the checking, the program will start the searching function to find whether the input name exists or not. If the input is not suitable for any record in the phone book, invalid data will show and ask user whether try again or not. All of the three-searching procedure will have the data check.

## 6. The Operations of Each Module

In this program, there are some main module in this program. They are the searching module, editing module, adding module, delete module and sorting module.

First, in the searching module, it consists three searching methods in this module. The user can determine to use which methods in their searching. In the searching name module, If the input data passes the data check, the searching algorithms will operate and search for the phone record. When the data pass the above two checking, the phone record, like the phone number and the email address. After all the process, the question try again will be asked. The whole module will be ended when the input is not satisfied with Y or y. In the searching phone and searching email modules apply the same searching and the data checking.

Next, in the editing module, it will be another important module in this program because the phone book user always need to modify the record of the phone book. So, in this module, the data check used in the name searching module will be reused in this module. After the data checking, the program will find out the existence of the input data. When the data is found, then the program ask for updating the phone

book record. When the user finished editing the phone record, the question "try again "will be asked.

Then, the deleting program will perform similar with the editing program. Only the statement output will be changed. As the data record is deleted, the file and the program will only displace some spaces and the blank column to represent the empty data.

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

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

## 7. The Sorting Algorithm

In this program, the sorting algorithm is used. In this phone book, I tend to use the bubble sort. As the phone book is always list according to the capital. So, using the 16-bubble sort can help to group the name record into the correct order. In this sorting, the target group is the names of the friends and others. So, the whole sorting will be carried out under the name. It is because people always use the name as the main searching category. So that, I think sorting name in the correct order will be convenient for the general user because the user doesn't need to adapt the unfamiliar searching interface when they are using this phone book system. Therefore, I believe that using the name as the sorting target is user-friendly and efficient for the user to use. Besides that, in this program, I also used the liner searching to search for the phone book data. It is because using the Liner Searching will be faster than the

Binary Searching. As the common phone book record is in a small scale, the liner searching is affordable in this kind of situation. Compare with the liner searching and binary searching, the speed of searching the data within the small scale of data is nearly the same. So, there is no an absolute advantage of using the binary searching in this program. Therefore, I think that using the liner searching in this phone book is reasonable and feasible.

## 8. User Guide of the Program

In this program, there are some variable in this program to facilitate the operation of the program. In this program, the array of Name[i], Phone[i] and Email[i] are used for storing the data of the phone record. On the other hand, the variable [i] is used as the counter in these arrays.

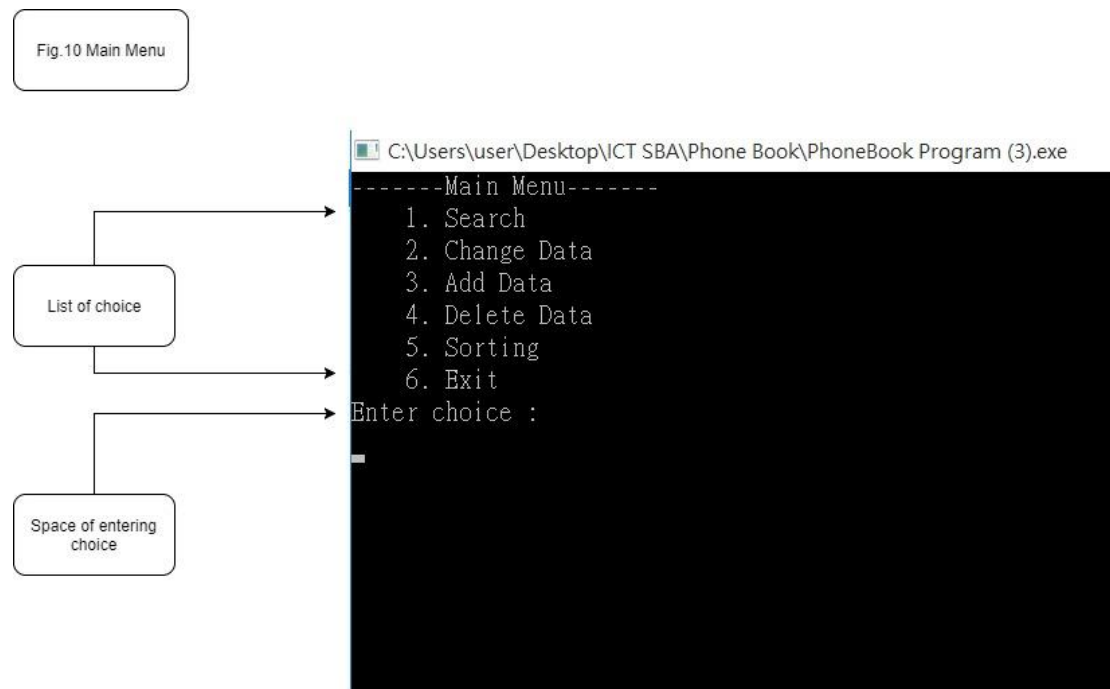




Fig.11 Searching Name

List of result

"Y","y" search again  
"X","x" Stop searching

Enter the searching  
name

```
C:\Users\user\Desktop\ICT SBA\Phone Book\PhoneBook Program (3).exe
Please enter the name : Chen Shao Gu
Name : Chen Shao Gu
Phone no. : 96033592
Email : djvuefmvja@nwihe.cn
Finished, Try again? (Y/N):N
Press <Enter> to go back the Main Menu.
```

Fig.12 Editing Data

List of result

"Y","y" edit again  
"X","x" Stop editing

Enter the editing  
name

```
C:\Users\user\Desktop\ICT SBA\Phone Book\PhoneBook Program (3).exe
Please enter the name : Chen Shao Gu
The phone no. of Chen Shao Gu is 96033592
Please enter the new phone no. : 75839578
The email address of Chen Shao Gu is djvuefmvja@nwihe.cn
Please enter the new email address: fbjehvjss@vndfjb.cn
Finished, Try again? (Y/N):N
Press <Enter> to go back the Main Menu.
```



# Testing and Evaluation

## 1. Test the Program

Fig.15 Test No.1

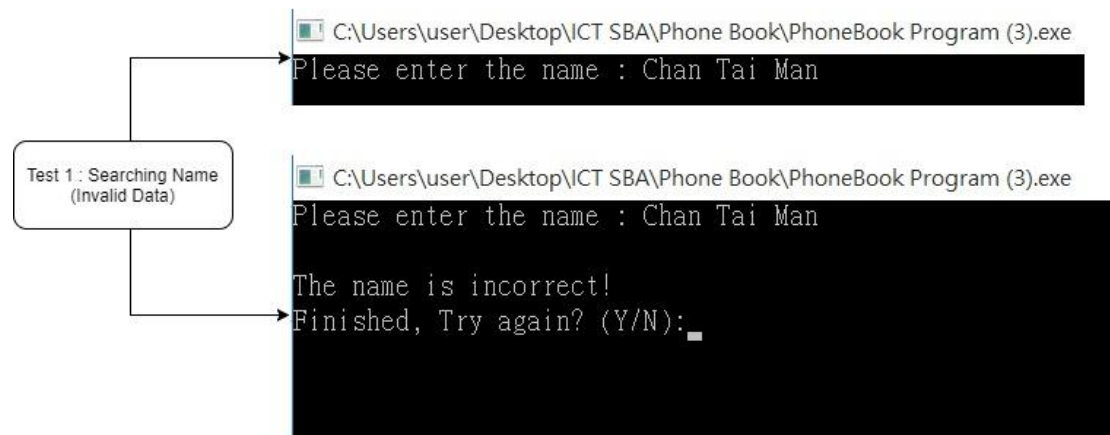


Fig.16 Test 2

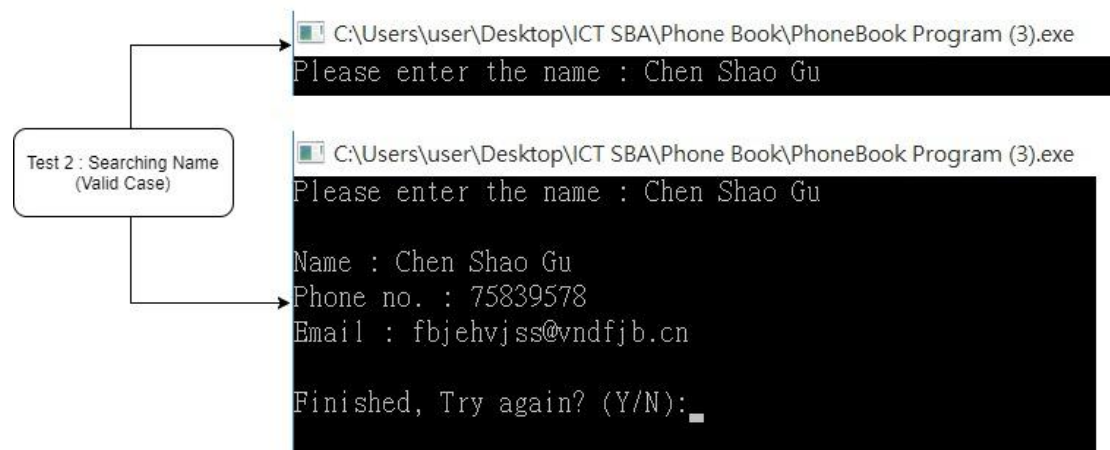


Fig.17 Test 3

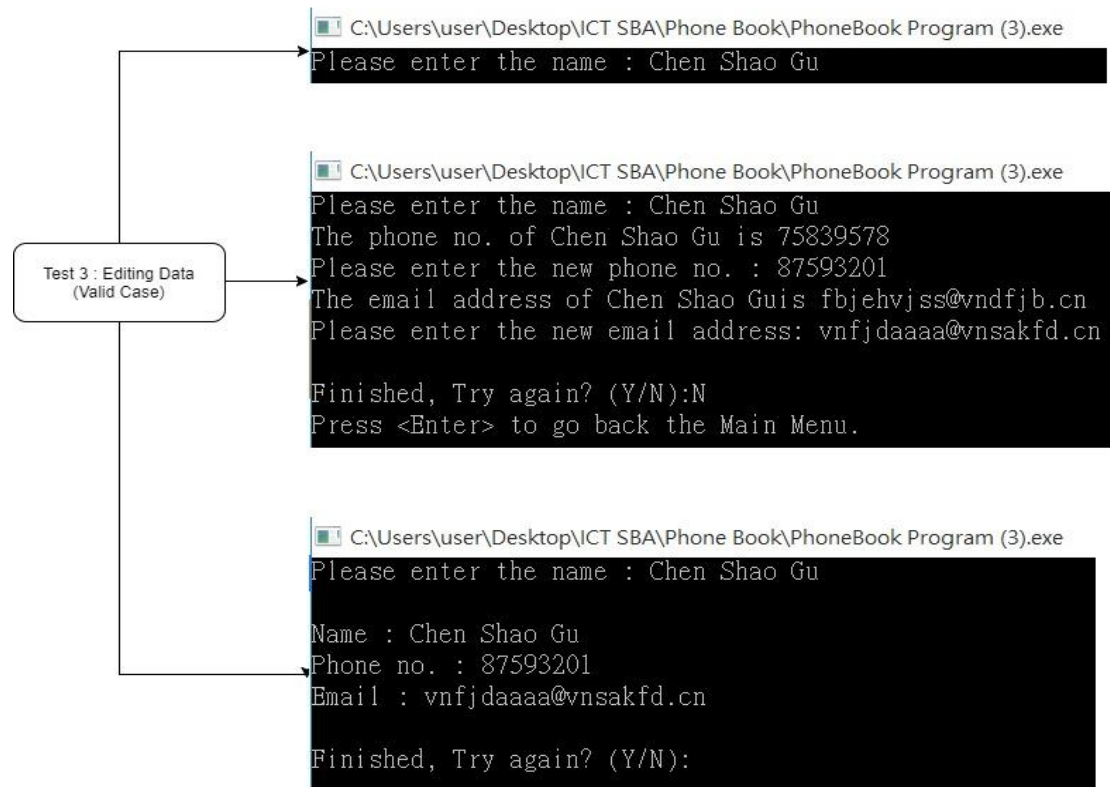


Fig.18 Test 4

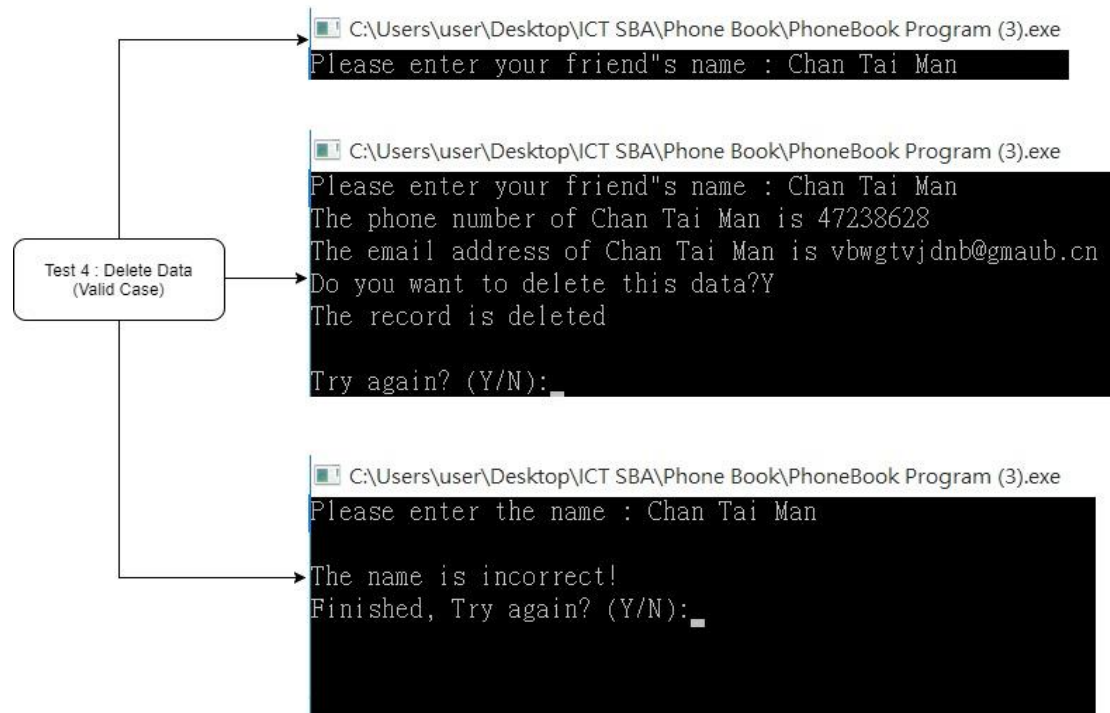


Fig.19 Test 5



### 3. Evaluate the Program

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

# Conclusion and Discussion

## 1. Conclusion

After finished the whole program, I think the program has meet the target and the aims of the project. In this program, it has provided a user-friendly interface. Although the interface seems well-performed, some data collection inside the program still have some space to improve. For me, it is the space that I can done better. Also, the content of the program could be clearer that can suitable for anyone of the computer user.

## 2. Skills Obtained from the Project

After finished the project, I have known more about program design and the program writing. It is because the new program creating is a tough work for me as there are many details need to correct during writing the program. After doing this program, I have developed the design skills and the writing skills of the program because I have to consider the user interface and the user-friendly environment. These difficulties have strengthened my program skills.

## 3. Difficulties

When I was doing the program, I always found a difficult that I often don't know how to loop the program as the phone book system need to do the same progress for many times. To comment with, I used the repeat loop and for loop to do the program because the repeat loop does not contain some restrict and the condition contain more freedom. As a result, after finished the project, I have learnt more about the use of looping and undertint the iteration of the program.

## 4. Future Development

In the Future, I want to improve my program skills more deeply. It is because my program is not as same as a real phone book system. It is not user-friendly enough. So that, In the future, I want to develop more about the program in terms of more take care about the users. Also, in the future, the user-interface will be planned to be changed because the

technology of the program is still developing. If I want to improve, I must catch up the development of the world. As a result, I am going to improve my skills of program to handle the tasks in the future.



# Project Management

## Activity Schedule

Date	Task to be done
May-2017	Choice of Topic, Background research + Define the objectives + Propose Functions
July-2017	Design of Solution
Oct-2017	Implementation
Nov-2017	Testing & Evaluation
Dec-2017	Final Report

# Appendix

1. ICT Textbooks Program Development (D1, D2)
2. ICT Textbooks Compulsory 3
3. Subjection Information on SBA

[http://www.hkeaa.edu.hk/en/sba/sub\\_info\\_sba/dse\\_subject.html?16&5](http://www.hkeaa.edu.hk/en/sba/sub_info_sba/dse_subject.html?16&5)

4. Wikipedia Dev-Pascal

[https://en.wikipedia.org/wiki/Pascal\\_%28programming\\_language%29](https://en.wikipedia.org/wiki/Pascal_%28programming_language%29)

5. Wikipedia Pascal Telephone Directory

[https://en.wikipedia.org/wiki/Telephone\\_directory](https://en.wikipedia.org/wiki/Telephone_directory)

## Source Code

```
program phonebook;

Uses Crt;

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

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

Procedure ShowData;
var i : integer;
begin
    writeln('====Data List====');
    for i := 1 to N do
    begin
        writeln('-----');
        writeln('Name : ', Name[i]);
        writeln('Phone no. : ', Phone[i]);
        writeln('Email : ', Email[i])
    end;
    writeln('-----')
```

end;

procedure ReadFile;

var f : text;

    i : integer;

begin

    assign(f, 'phone.txt');

    reset(f);

    i := 0;

    while not eof(f) do

        begin

            i := i + 1;

            readln(f, Name[i]);

            readln(f, Phone[i]);

            readln(f, Email[i]);

        end;

    close(f);

    N := i

end;

procedure WriteFile;

var Outfile : text;

    i : integer;

begin

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

```
procedure SearchName;  
var i : integer;  
    phname : string;  
    found : Boolean;  
    choice : char;  
begin  
    repeat  
        Clrscr;  
        repeat  
            write('Please enter the name : ');  
            readln(phname);  
        until phname <> '';
```

```

writeln;

found := false;

i := 0;

repeat
    i := i + 1 ;

    {for j := 1 to (length(name[i]) - length(phname) + 1) do}

    if phname = Name[i] then

        found := true

until (found) or (i = N);

if found then

    begin

        writeln('Name : ', Name[i]);

        writeln('Phone no. : ', Phone[i]);

        writeln('Email : ', Email[i]);

        writeln

    end

    else writeln('The name is incorrect!');

write('Finished, Try again? (Y/N):');

readln(choice);

until choice in ['N', 'n'];

writeln('Press <Enter> to go back the Main Menu.');
```

WriteFile;

```

end;
```

```

procedure SearchPhoneNo;
var i : integer;
    number : string;
    found : Boolean;
    choice : char;
begin
repeat
Clrscr;
repeat
    write('Please enter the phone no. : ');
    readln(number);
until number <> "";
writeln;
found := false;
i := 0;
repeat
    i := i + 1 ;
    {for j := 1 to (length(Phone[i]) - length(number) + 1) do}
        if number = Phone[i] then
            found := true
until (found) or (i = N);
    if found then
        begin
            writeln('Name : ', Name[i]);

```

```

        writeln('Phone no. : ', Phone[i]);
        writeln('Email : ', Email[i]);
        writeln
    end
    else writeln('The phone no. is incorrect!');

write('Finished, Try again? (Y/N):');
readln(choice);
until choice in ['N', 'n'];
writeln('Press <Enter> to go back the Main Menu. ');
WriteFile;
end;

```

```

procedure SearchEmail;
var i : integer;
    email : string;
    found : Boolean;
    choice : char;
begin
repeat
Clrscr;
repeat
    write('Please enter the email address : ');
    readln(email);

```



```

until email <> "";
writeln;
found := false;
i := 0;
repeat
    i := i + 1 ;
    {for j := 1 to (length(Email[i]) - length(email) + 1) do}
        if email = Email[i] then
            found := true
until (found) or (i = N);
if found then
    begin
        writeln('Name : ', Name[i]);
        writeln('Phone no. : ', Phone[i]);
        writeln('Email : ', Email[i]);
        writeln
    end
    else writeln('The email address is incorrect!');

write('Finished, Try again? (Y/N):');
readln(choice);
until choice in ['N', 'n'];
writeln('Press <Enter> to go back the Main Menu. ');
WriteFile;
end;

```

```

procedure Searching;
var
    choice : integer ;
begin
repeat
Clrscr;

    writeln('      Search Menu              ');
    writeln;
    writeln('  1. Searching name              ');
    writeln('  2. Searching phone no.          ');
    writeln('  3. Searching email address      ');
    writeln('  4. Quit                        ');
    writeln;
    write(' Enter choice: ');
    readln(Choice);
    writeln;
    case Choice of
        1 : SearchName;
        2 : SearchPhoneNo;
        3 : SearchEmail;
    end;
readln;

```

```
until choice = 4;

writeln('Press <Enter> to go back the Main Menu. ');

end;
```

```
procedure ChangeData;

var i : integer;

    phname : string;

    valid : Boolean;

    choice : char;

begin

repeat

Clrscr;

repeat

    write('Please enter the name : ');

    readln(phname);

    until phname <> "";

    valid := false;

    i := 0;

    repeat

        i := i + 1;

        if Name[i] = phname then

            valid := true
```

```

        until (valid) or (i >= N);
        if valid then
            begin
                writeln('The phone no. of ', Name[i] , ' is ',
Phone[i]);

                write('Please enter the new phone no. : ');
                readln(Phone[i]);

                writeln('The email address of ', Name[i] , 'is ',
Email[i]);

                write('Please enter the new email address: ');
                readln(Email[i]);

                writeln

            end
        else writeln('The name is not exist!');

```

```

write('Finished, Try again? (Y/N):');
readln(choice);
until choice in ['N', 'n'];
writeln('Press <Enter> to go back the Main Menu. ');
WriteFile;
end;

```

```

procedure AddData;
var newname, newphoneno, newemail : string;

```

```
        i : integer;
        choice : char;
begin
    Clrscr;
    repeat
        i := 0;
        write('Please enter the new name : ');
        readln(newname);
        write('Please enter the new phone no. : ');
        readln(newphoneno);
        write('Please enter the new email address : ');
        readln(newemail);
        N := N + 1;
        Name[i] := upcase(newname);
        Phone[i] := newphoneno;
        Email[i] := newemail;
        write('Data added, try again? (Y/N):');
        readln(choice);
    until choice in ['N', 'n'];
    writeln('Press <Enter> to go back the Main Menu. ');
    WriteFile;
end;
```

```

procedure DeleteData;
var i : integer;
    valid : Boolean;
    fdname : string;
    choice : char;
begin
repeat
Clrscr;
    repeat
write('Please enter your friend"s name : ');
readln(fdname);
until fdname <> "";
    valid := false;
    i := 0;
    repeat
i := i + 1;
        if Name[i] = fdname then
            valid := true;
    until (valid) or (i >= N);
    if valid then
        begin
            writeln('The phone number of ', Name[i] , ' is ', Phone[i]);
            writeln('The email address of ', Name[i] , ' is ', Email[i]);
            write('Do you want to delete this data?');
            readln(choice);

```

```
    if choice in ['Y','y'] then
        begin
            Name[i] := "";
            Phone[i] := "";
            Email[i] := "";
            writeln('The record is deleted');
            writeln
        end
    else writeln('The name is not exist!');
end;
```

```
write('Try again? (Y/N):');
readln(choice);
until choice in ['N', 'n'];
writeln('Press <Enter> to go back the Main Menu. ');
WriteFile;
end;
```

```
procedure Sorting;
var
    i , j : integer;
    phname, phoneno, emailaddress : string;
    { swap : boolean;}
begin
```

```
Clrscr;

j := 0;

{swap := false; }

repeat

j := j + 1;

for i := 1 to N - j do

if Name[i] > Name[i + 1] then

begin

    phname := upcase(Name[i]);

    Name[i] := Name[i + 1];

    Name[i + 1] := phname;

    phoneno := Phone[i];

    Phone[i] := Phone[i + 1];

    Phone[i + 1] := phoneno;

    emailaddress := Email[i];

    Email[i] := Email[i + 1];

    Email[i + 1] := emailaddress;


    {swap := true;}

end;

until (j = N - 1) {or not swap};

ShowData;

WriteFile;

end;
```



```
begin { Main Program }  
    ReadFile;  
    Clrscr;  
    repeat  
        writeln('-----Main Menu-----');  
        writeln('    1. Search          ');  
        writeln('    2. Change Data      ');  
        writeln('    3. Add Data         ');  
        writeln('    4. Delete Data      ');  
        writeln('    5. Sorting          ');  
        writeln('    6. Exit             ');  
        writeln('Enter choice : ');  
        readln(choice);  
        writeln;  
        case choice of  
            1 : Searching;  
            2 : ChangeData;  
            3 : AddData;  
            4 : DeleteData;  
            5 : Sorting;  
        end;  
    end;
```

```
    readln  
until choice = 6;  
end.
```