Hong Kong Diploma of Secondary Education Examination 20XX

Information and Communication Technology

(Coursework)

Option D: Software Development

Title: Subject Selection System

Cheung Sha Wan Catholic Secondary School

Candidate ID: XXXXX

Chapter 1 Introduction

1.1 Situation & Background

Under the New Senior Secondary (NSS) curriculum, students are required to select elective subjects from among the various subjects offered by schools. There are various constraints on the allocation of subjects to students.

Suppose you are the IT project manager responsible designing a student-subject allocation system for a secondary school. The system should provide functions that facilitate data management for student-subject allocation, such as:

- management of student personal records
- management of selection and allocation constraints
- inputting of elective preference
- reporting of allocation result

1.2 Objectives

I am you are the IT project manager responsible designing a student-subject allocation system for a secondary school. I need to design a program that aims at secondary school students for the elective selection use. The program should consist of different parts and with different functions including login system, subject display, subject selection, selection display and password changing. The program also needs to record the choices of students and save the selection in students' personal records.

I aim at designing a clear and user-friendly program so that the users need not learn how to use the program and so to increase the efficiency of the whole process. The program is cost effective as it needs not a large amount of resources for maintenance and the system requirement is low so that almost every computer can run it. As the program is light and is in small size, users need not worry about the program will use a lot of resources of computer.

Chapter 2 Design

2.1 Brief Description

In this chapter, I will design the structure of my program based on the functions mentioned in Chapter 1. I will design the process and rundown of my program and the data structure of the text file of the 3 elective blocks and students' personal records including User ID, password, name and elective selection. And also I will design the interface and menu for the users. There are several functions in my program including login before use, display students' choices, choose electives and change password. Every change will be saved in the corresponding text files after execution of function automatically.

There will be assumption and some rules about the program:

- The maximum number of students is 100 as the constant value
- Each student only can select totally 3 subjects in different blocks
- Each student only can select 1 subject from certain block
- Students cannot choose duplicate subjects. E.g. both 1st and 3rd electives are Physics
- Students cannot choose the subject if the quota of the subject is zero
- Students need to use their user ID and password to login the system for further process

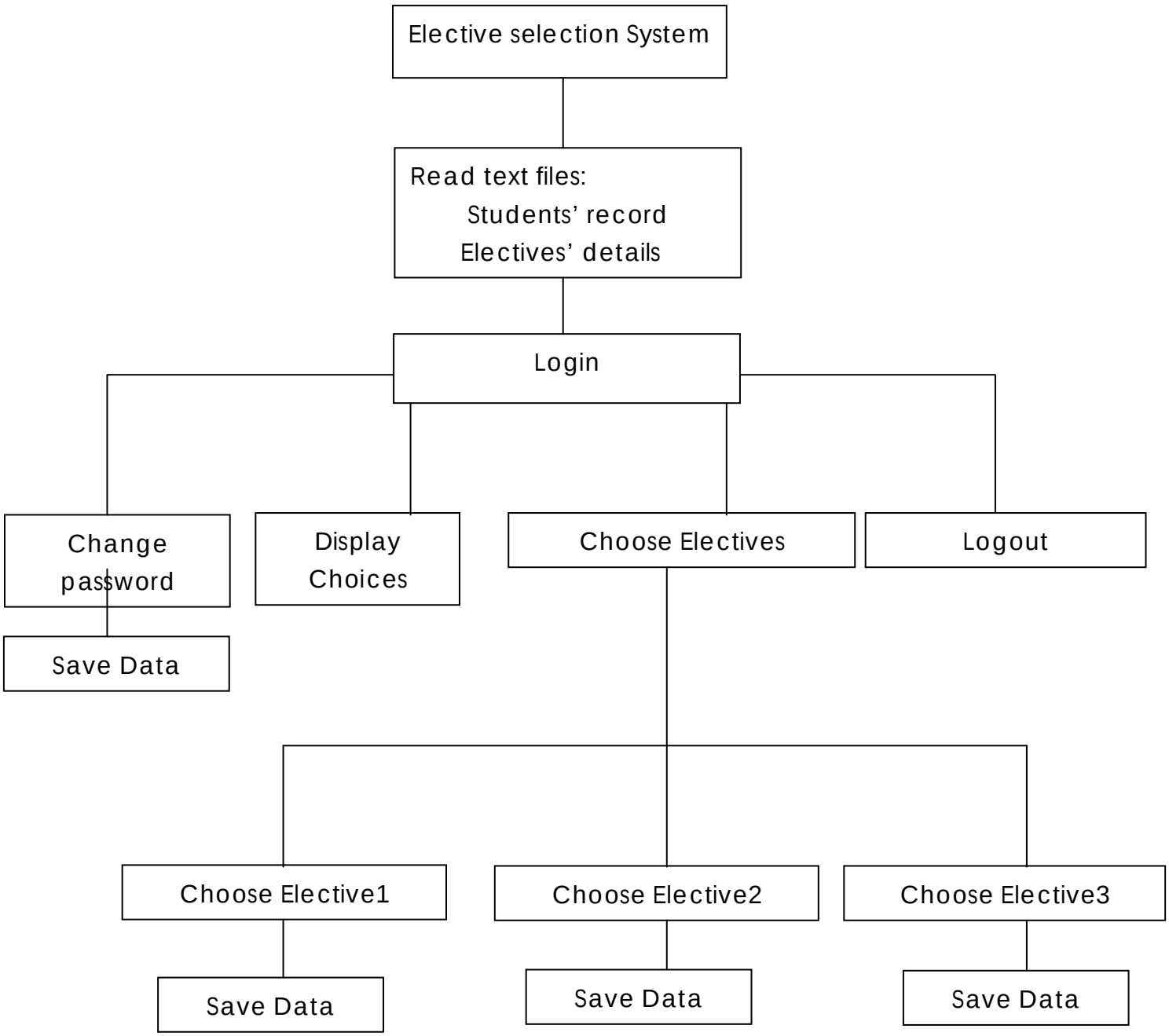
2.2 Refinement of problems

To make a program that can provide different functions, the program should be divided into different parts to prevent from chaos due to the messy codes. To make the program more simple and easy to understand, I had divided it into different parts as following:

- Reading text files from the file
- Login
- Displaying Saved Data
- Changing Password
- Choosing Elective1
- Choosing Elective2
- Choosing Elective3
- Saving Changes to the text files
- Logout

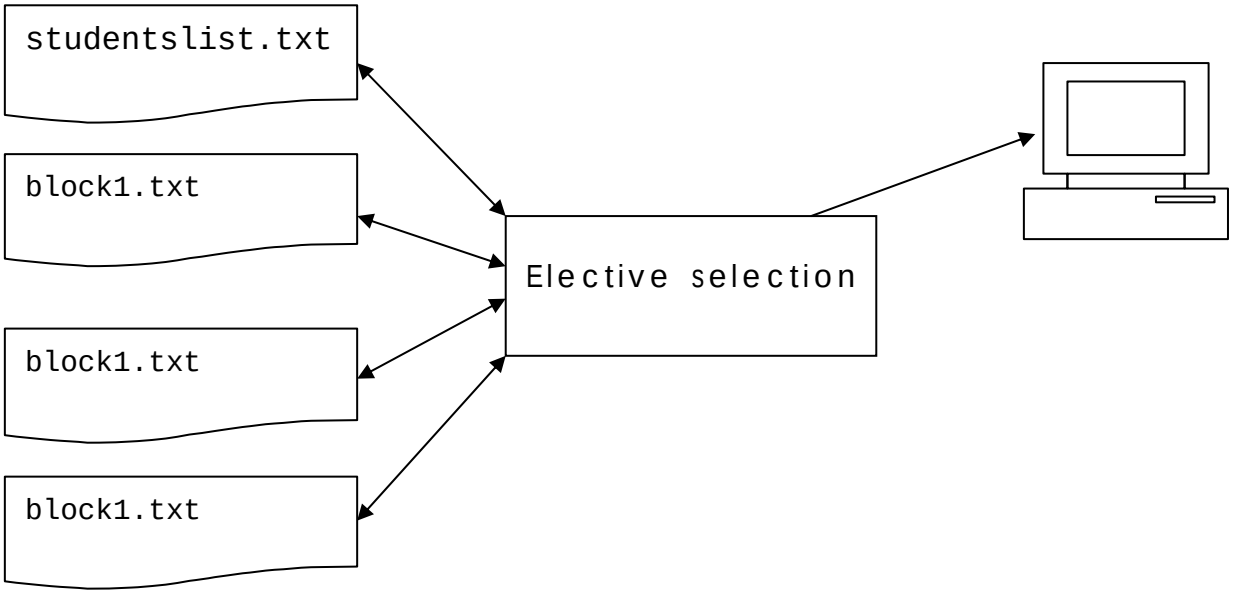
In my program, students need to login for the system before carry out any of the functions. Students need to use their own user ID and password to login the system for further actions. After successfully login to the system, my program will display a menu for users to input their choice referred to the correspond actions. Each elective selection is independent with other elective selections. It means that my program will not force students to make all decisions on the 3 blocks of electives. It has reduced the limitation of the program to the users.

The problems can be divided as the following chart:



Data Flow:

1. The program will read 4 text files in the same folder
    i. studentslist.txt—storing students' ID, passwords, names and selected electives
    ii. block1.txt —storing the block1 electives' index, name and quotas
    iii. block2.txt —storing the block2 electives' index, name and quotas
    iv. block3.txt —storing the block3 electives' index, name and quotas
2. The program will display the corresponding content under commands of the users
3. The program will make changes and save to the 4 files after the users have finished choosing any of electives or changing passwords



2.3 Input Data File Formats

There are 4 text files in my program.

Studentslist.txt:

- This text file store the User ID of the students, the passwords , names and the 3 blocks of elective selections
- The file is used to as an indicator of the login part of the program to check if the users input correct values
- The text file also saves the choices of the students on elective selection
- The program will use the content of this file to display the data of the students including student no. , name, electives.
- Here is the format of the data
  - Student number/User ID :
    - String type
    - s+3 numbers E.g. s001
  - Password :
    - String type
    - Default password=1234
  - Name
    - string
  - Elective 1
    - string
  - Elective 2
    - string
  - Elective 3
    - string

E.g

s001	→ Student number/User ID
1234	→ Password
Au Sham Ki, Bobby	→Name
Chemistry	→Elective1
ICT	→Elective2

PE	→Elective3

Sample Files:

Studentslist.txt

```
s001
1234
Au Sham Ki, Bobby
Chemistry
ICT
PE
s002
1234
Au Yue, Joanne
Economics
Geography
Chemistry
s003
1235
Chan Kai Bong
Economics
……................
……..................
……..................
```

Store the personal data of the 1st student

Store the personal data of the 2nd student

block1.txt:

- this file stores the block 1 electives
  - including elective index as an indicator for selection
  - elective name to display for the users and save to studentslist.txt
  - number of quota of different electives
- type of the record
  - elective index: integer
    - for 1-3
  - elective name : string
  - elective quota : integer
    - for 0-30

E.g.

1	->Elective index
Chemistry	->Elective name
29	->Elective quota

Sample File:

block1.txt

1
Chemistry
29

Store the data of the 1st subject in block1

2
Physics
28

Store the data of the 2nd subject in block1

3
Economics
28

Store the data of the 3rd subject in block1

block2.txt:

- this file stores the block 2 electives
  - including elective index as an indicator for selection
  - elective name to display for the users and save to studentslist.txt
  - number of quota of different electives
- type of the record
  - elective index: integer
    - for 1-7
  - elective name : string
  - elective quota : integer
    - for 0-30

E.g.

1	→Elective index
ICT	→Elective name
18	→Elective quota

Sample File:

block2.txt

1
ICT
18
} Store the data of the 1st subject in block2

2
Chemistry
30
} Store the data of the 2nd subject in block2

3
Physics
30
} Store the data of the 3rd subject in block2

4
Biology
29
} Store the data of the 4th subject in block2

……

block3.txt:

- this file stores the block 3 electives
  - including elective index as an indicator for selection
  - elective name to display for the users and save to studentslist.txt
  - number of quota of different electives
- type of the record
  - elective index: integer
    - for 1-5
  - elective name : string
  - elective quota : integer
    - for 0-30

E.g.

2	->Elective index
Physics	->Elective name
30	->Elective quota

Sample File:

block3.txt

1
Chemistry
29
} Store the data of the 1st subject in block3

2
Physics
30
} Store the data of the 2nd subject in block3

3
Biology
30
} Store the data of the 3rd subject in block3

4
PE
19
} Store the data of the 4th subject in block3

……

2.4 Output Data Format

My program does not have an independent text file for display and output the data. The output data format is according to the 4 text files mentioned above. My program does not have an extra file for storing the saved data but use the original text files. My program can obtain the data from the 4 text files and display in the result pages of the program.

There are different data to be output in my program:

Display data page

1. Student id
2. Student name
3. Elective 1 of the student (RESULT)
4. Elective 2 of the student (RESULT)
5. Elective 3 of the student (RESULT)

Sample layout

```
[ ELECTIVE   SELECTION   SYSTEM ]

<><><><><><><><><><><><><><><><><><><><><><><><>

[STUDENT ID ]   : s001

[     NAME       ]   : Au Sham Ki, Bobby


[ ELECTIVE1 ]   : Chemistry

[ ELECTIVE2 ]   :   ICT

[ ELECTIVE3 ]   :   PE

<><><><><><><><><><><><><><><><><><><><><><><><>

<<< Press <Enter> to return. >>>
```

In Elective1 choosing page

1. Elective 1 Index
2. Elective 1 Name
3. Elective 1 Quota

Sample layout:

```
 [ELECTIVE1 SELECTION]

Choice                  Elective                 Quota
<><><><><><><><><><><><><><><><><><><><><><><><><><>

1                     Chemistry                29
2                     Physics                  28
3                     Economics                28

<><><><><><><><><><><><><><><><><><><><><><><><><><>

Please Choose Your Elective1
(1-3, Others To Reset Your Choice )
Enter Your Choice:
```

In Elective2 choosing page

1. Elective 2 Index
2. Elective 2 Name
3. Elective 2 Quota

Sample layout:

```
[ELECTIVE2 SELECTION]

Choice                  Elective                  Quota
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

1                       ICT                       18
2                       Chemistry                 30
3                       Physics                   30
4                       Biology                   29
5                       Geography                 19
6                       PE                        20
7                       VA                        20
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>

Please Choose Your Elective2
(1-7, Others To Reset Your Choice )
Enter Your Choice:
```

In Elective3 choosing page

1. Elective 3 Index
2. Elective 3 Name
3. Elective 3 Quota

Sample layout

[ELECTIVE3 SELECTION]

Choice	Elective	Quota
1	Chemistry	29
2	Physics	30
3	Biology	30
4	PE	19
5	Economics	29

Please Choose Your Elective3
(1-5, Others To Reset Your Choice)
Enter Your Choice:

Chapter 3 Implementation

3.1 Brief Description

In this chapter, I will discuss the implementation of the Subject selection program. I will discuss the data structures in the program, the procedures in the program with brief description about the functions, explain the main algorithms used in the program, display the program codes and display the user interface.

3.2Data Structure

The following parallel array will be used to store the student id, student password, student name, elective1, elective2 and elective3 respectively in the text file of studentslist.txt.

studid : array[1..max_stud] of string

studpw : array[1..max_stud] of string

studname : array[1..max_stud] of string

elective1 : array[1..max_stud] of string

elective2 : array[1..max_stud] of string

elective3 : array[1..max_stud] of string

where max_stud is a constant defined at the beginning of the program.

s001
1234
Au Sham Ki, Bobby
Physics
Chemistry
Economics
s002
1234
Au Yue, Joanne

studid	studpw	studname
s001	1234	Au Sham Ki, Bobby

elective1	elective2	elective3
Physics	Chemistry	Economics

The following parallel array will be used to store the elective1 index, elective1 name and elective1 quota respectively in the text file of block1.txt.

e1_index:array[1..max_sub] of integer;

e1:array[1..max_sub] of string;

quota_e1: array[1..max_sub] of integer;

where max_sub is a constant defined at the beginning of the program.

```
1
Chemistry
30
2
Physics
27
3
Economics
28
```

e1_index	e1	quota_e1
1	Chemistry	30
2	Physics	27

The following parallel array will be used to store the elective1 index, elective1 name and elective1 quota respectively in the text file of block1.txt.

e2_index:array[1..max_sub] of integer;

e2:array[1..max_sub] of string;

quota_e2: array[1..max_sub] of integer;

where max_sub is a constant defined at the beginning of the program.

```
1
ICT
19
2
Chemistry
29
3
Physics
30
4
```

e2_index	e2	quota_e2
1	ICT	19
2	Chemistry	29

The following parallel array will be used to store the elective1 index, elective1 name and elective1 quota respectively in the text file of block1.txt.

e3_index:array[1..max_sub] of integer;

e3:array[1..max_sub] of string;

quota_e3: array[1..max_sub] of integer;

where max_sub is a constant defined at the beginning of the program.

```
1
Chemistry
29
2
Physics
30
3
Biology
30
4
```

e3_index	e3	quota_e3
1	Chemistry	29
2	Physics	30

3.3 Procedures

The following procedures will be used in the program construction

1. read_students

   This procedure is used to read the text file of 'studentslist.txt'

```
  procedure read_students;
var
  i : integer;
  studlist : text;
begin
  assign(studlist, 'studentslist.txt');
  reset(studlist);
  i := 0;
  while not eof(studlist) do
    begin
      i := i + 1;
      readln(studlist, studid[i]);
      readln(studlist, studpw[i]);
      readln(studlist, studname[i]);
      readln(studlist, elective1[i]);
      readln(studlist, elective2[i]);
      readln(studlist, elective3[i]);
    end;
  num_stud := i;
  close(studlist)
end;
```

2. read_e1

   This procedure is used to read the text file of 'block1.txt'

```
  procedure read_e1;
var
  i : integer;
  block1 : text;
begin
  assign(block1, 'block1.txt');
  reset(block1);
  i := 0;
  while not eof(block1) do
    begin
      i := i + 1;
      readln(block1, e1_index[i]);
      readln(block1, e1[i]);
      readln(block1, quota_e1[i]);

    end;
  num_e1 := i;
  close(block1)
end;
```

3. read_e2;

This procedure is used to read the text file of 'block2.txt'

```
procedure read_e2;
var
  i : integer;
  block2 : text;
begin
  assign(block2, 'block2.txt');
  reset(block2);
  i := 0;
  while not eof(block2) do
    begin
      i := i + 1;
      readln(block2, e2_index[i]);
      readln(block2, e2[i]);
      readln(block2, quota_e2[i]);

    end;
  num_e2 := i;
  close(block2)
end;
```

4. read_e3;

This procedure is used to read the text file of 'block3.txt'

```
procedure read_e3;
var
  i : integer;
  block3 : text;
begin
  assign(block3, 'block3.txt');
  reset(block3);
  i := 0;
  while not eof(block3) do
    begin
      i := i + 1;
      readln(block3, e3_index[i]);
      readln(block3, e3[i]);
      readln(block3, quota_e3[i]);

    end;
  num_e3 := i;
  close(block3)
end;
```

5. login

This procedure is used to create a login interface and login function to the users.

Users can login to the system with their user id (studid) and password(studpw).

This procedure will check the id and password if they are correct and use function Getpword to hide the password with asterisks(*) .

```
procedure login(var stud_index : integer);
var
  userid, password : string;
  found : boolean;
  i : integer;
begin
  clrscr;
  writeln;
  writeln;
  textcolor(white);
  writeln('[CHEUNG SHA WAN CATHOLIC SECONDARY SCHOOL]');
  writeln('');
  writeln('[SENOIR FORM STUDENT]');
  writeln('');
  writeln('[ELECTIVE]');
  writeln('');
  writeln('[SELECTION]');
  writeln('');
  writeln('[SYSTEM]');
  textcolor(white);
  writeln;
  writeln;
  writeln('[LOGIN]');
  writeln;
  write('[   ID   ] : ');
```

```
  readln(userid);
  writeln;
  write('[Password] : ');
  password := GetPword;
  writeln;
  found := false;
  i := 0;
  while (i < num_stud) and (not found) do
    begin
      i := i + 1;
      if (userid = studid[i]) and (password = studpw[i]) then
        begin
          found := true;
          stud_index := i
        end
    end;
  if not found then
    begin
      stud_index := 0;
      textcolor(6);
      writeln('<<<Invalid UserID or Password>>>');
      writeln;
      textcolor(white);
      write('<<<Press <Enter> to refresh>>>');
      readln
    end;
end;
```

6. GetPWord

Ref:
http://computer-programming-forum.com/29-pascal/7af4f3f05f738777.htm

It is a function for hiding the input password to increase the security. The character typed in password column will return in the same length of asterisks.

```
{ Ref: http://computer-programming-forum.com/29-pascal/7af4f3f05f738777.htm }
function GetPWord : string;    (* A function for hiding password *)
var
  S : string;
  C : Char;
begin
  S := '';
  repeat
    C := ReadKey;
    if (C <> #10) and (C <> #13) and (C <> #8) then
      begin
        S := S + C;
        write('*');
      end
    else if C = #8 then
      begin
        S[0] := Chr(Length(S) - 1);
        GotoXY(WhereX - 1, WhereY);
        write(' ');
        GotoXY(WhereX - 1, WhereY);
      end;
  until (C = #10) or (C = #13);
  GetPWord := S;
  writeLn;
end;
```

7. store_students

After finishing every edition and input of the users, the text file will be rewritten. This procedure is used to rewrite the content of the text file of 'studentslist.txt'

```
   procedure store_students;
var  i : integer;
     studlist : text;
begin
  assign(studlist, 'studentslist.txt');
  rewrite(studlist);
  for i := 1 to num_stud do
    begin
      writeln(studlist, studid[i]);
      writeln(studlist, studpw[i]);
      writeln(studlist, studname[i]);
      writeln(studlist, elective1[i]);
      writeln(studlist, elective2[i]);
      writeln(studlist, elective3[i]);
    end;
  close(studlist)
end;
```

8. store_subject1 store_subject2 store_subject3

After finishing every edition and input of the users, the text file will be rewritten. This procedure is used to rewrite the content of the text files of ‘block1.txt’, ‘block2.txt’, ‘block3.txt’ respectively.

```
 procedure  store_subject1;
var  i : integer;
     block1 : text;
begin
  assign(block1, 'block1.txt');
  rewrite(block1);
  for i := 1 to num_e1 do
    begin
      writeln(block1, e1_index[i]);
      writeln(block1, e1[i]);
      writeln(block1, quota_e1[i]);
    end;
  close(block1)
end;
```

```
  procedure  store_subject2;
var  i : integer;
     block2 : text;
begin
  assign(block2, 'block2.txt');
  rewrite(block2);
  for i := 1 to num_e2 do
    begin
      writeln(block2, e2_index[i]);
      writeln(block2, e2[i]);
      writeln(block2, quota_e2[i]);
    end;
  close(block2)
end;
```

```
 procedure  store_subject3;
var  i : integer;
     block3 : text;
begin
  assign(block3, 'block3.txt');
  rewrite(block3);
  for i := 1 to num_e3 do
    begin
      writeln(block3, e3_index[i]);
      writeln(block3, e3[i]);
      writeln(block3, quota_e3[i]);
    end;
  close(block3)
end;
```

9. main_menu

This procedure is used to provide a user interface for user to carry out different function after the successfully login to the system.

```
procedure main_menu(stud_index : integer);
var
  choice : integer;
begin
  repeat
    clrscr;
    writeln;
    writeln('[ELECTIVE SELECTION SYSTEM]');
    writeln;
    writeln('[MAIN MENU]');
    writeln;
    writeln('<><><><><><><><><><><><><><><><><><><><><>');
    writeln;
    writeln('[1] Display Chioces');
    writeln('[2] Change password');
    writeln('[3] Choose Elective1');
    writeln('[4] Choose Elective2');
    writeln('[5] Choose Elective3');
    writeln('[6] Quit');
    writeln;
    writeln('<><><><><><><><><><><><><><><><><><><><><>');
    writeln;
    write('Enter Choice: ');
    readln(Choice);
    writeln;
    case choice of

      1 : display(stud_index);
      2 : change_password(stud_index);
      3 : choose_e1(stud_index);
      4 : choose_e2(stud_index);
      5 : choose_e3(stud_index);
    end;
  until choice = 6;
end;
```

10. display

This procedure is used to display the data saved in the text file of 'studentslist.txt.' but not including student password (studpw) to the corresponding student.

```
procedure display(stud_index : integer);
begin
  clrscr;
  writeln;
  textcolor(white);
  writeln('[ ELECTIVE  SELECTION  SYSTEM ]');
  writeln;
  writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><> ');
  writeln;
  writeln('[STUDENT ID ]  : ', studid[stud_index]);
  writeln;
  writeln('[   NAME    ]  : ', studname[stud_index]);
  writeln;
  writeln;
  writeln;
  writeln('[ ELECTIVE1 ]  : ', elective1[stud_index]);
  writeln;
  writeln('[ ELECTIVE2 ]  : ', elective2[stud_index]);
  writeln;
  writeln('[ ELECTIVE3 ]  : ', elective3[stud_index]);
  writeln;
  writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><> ');
  writeln;
  textcolor(white);
  write('<<< Press <Enter> to return. >>>');
  readln ;

end;
```

11. change_password

This procedure is used to change the old password of the user. Users need to input the correct old password and enter the new password twice for confirmation. Incorrect input value will not be accept and students need input the data again

```
procedure change_password(stud_index : integer);
var
  oldpass, newpass1, newpass2 : string;
  pwchanged : boolean;
begin
  pwchanged := false;
  repeat
    clrscr;
    writeln;
    writeln('[CHANGE PASSWORD]');
    WRITELN;
    write('[Please Enter Your Old Password]       : ');
    oldpass := GetPword;
    if oldpass <> studpw[stud_index] then
      begin
        writeln;
        TEXTCOLOR(6);
        writeln('<<<Wrong Old Password>>>');
        TEXTCOLOR(WHITE);
        writeln;
        write('<<<Press <Enter> to retry>>>');
        readln
      end
    else
      begin
        writeln;
        write('[Please Enter Your New Password]       : ');
        newpass1 := GetPword;
        writeln;
        write('[Please Enter Your New Password Again] : ');
        newpass2 := GetPword;
        if newpass1 <> newpass2 then
```

12. choose_e1

This procedure is used to choose the elective1 of the students. It will display the array in the 'block1.txt' first and let the users input their choices. The choice is corresponding to the e1_index. For choice of 0 quotas, same subject as other elective block, the program will ask for input again until users input the valid value. Users can reset their choices by input the number out of the range of e1_index.

```
procedure choose_e1(var stud_index:integer);
var i,j,choice1:integer;
    choice_check:boolean;
begin
clrscr;
j:=find_sub_index1(elective1[stud_index]);
If j<>0 then
            quota_e1[j]:= quota_e1[j]+1;

writeln;
writeln('[ELECTIVE1 SELECTION]              ');
writeln;
writeln('Choice                Elective                Quota    ');
writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><><><> ');
writeln;
i:=1 ;
 for i:= i to num_e1
 do begin
         writeln('',e1_index[i]:3,'',e1[i]:28,'',quota_e1[i]:18);
    end  ;
writeln;
writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><><><> ');
repeat
textcolor(white);
writeln;
writeln('Please Choose Your Elective1') ;
writeln('(1-',num_e1,', Others To Reset Your Choice )');
write('Enter Your Choice: ');
readln(choice1);
writeln;
choice_check :=false;
textcolor(6);

      if (choice1>=1) and (choice1<=num_e1) then
         if ( quota_e1[choice1] >0)          then
             if  (e1[choice1]<>elective2[stud_index]) then
                 if  (e1[choice1]<>elective3[stud_index]) then
                        begin
                          choice_check:=true;
                          elective1[stud_index]:= e1[choice1];
                          quota_e1[choice1]:=quota_e1[choice1]-1;
                          textcolor(yellow);
                          writeln('<<<You have choosen ',e1[choice1],' as your Elective1>>>');
                          textcolor(white);
                        end
                 else writeln('<<<Duplicate choice>>>')
             else writeln('<<<Duplicate choice>>>')
         else writeln('<<<Not Enough Quota>>>')
      else
            begin choice_check:=true;
                  elective1[stud_index]:= e1[choice1];
                  textcolor(cyan);
                  writeln;
                 writeln('<<<You Have Reset Your Elective1>>>');
                 textcolor(white);
            end
until choice_check ;
store_students;
store_subject1;
textcolor(white);
writeln;
write('<<<Input <Enter> to Return>>>');
readln;
end;
```

13. choose_e2

This procedure is used to choose the elective1 of the students. It will display the array in the 'block2.txt' first and let the users input their choices. The choice is corresponding to the e2_index. For choice of 0 quotas, same subject as other elective block, the program will ask for input again until users input the valid value. Users can reset their choices by input the number out of the range of e2_index.

```
procedure choose_e2(var stud_index:integer);
var i,j,choice2:integer;
    choice_check:boolean;
begin
clrscr;
j:=find_sub_index2(elective2[stud_index]);
If j<>0 then
            quota_e2[j]:= quota_e2[j]+1;

writeln;
writeln('[ELECTIVE2 SELECTION]              ');
writeln;
writeln('Choice                Elective                 Quota    ');
writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><><><> ');
writeln;
i:=1 ;
 for i:= i to num_e2
 do begin
         writeln('',e2_index[i]:3,'',e2[i]:28,'',quota_e2[i]:18);
    end  ;
writeln;
writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><><><> ');
repeat
textcolor(white);
writeln;
writeln('Please Choose Your Elective2') ;
writeln('(1-',num_e2,', Others To Reset Your Choice )');
write('Enter Your Choice: ');
readln(choice2);
writeln;
choice_check :=false;
textcolor(6);

     if (choice2>=1) and (choice2<=num_e2) then
        if ( quota_e2[choice2] >0)          then
            if  (e2[choice2]<>elective1[stud_index]) then
                if  (e2[choice2]<>elective3[stud_index]) then
                       begin
                         choice_check:=true;
                         elective2[stud_index]:= e2[choice2];
                         quota_e2[choice2]:=quota_e2[choice2]-1;
                         textcolor(yellow);
                         writeln('<<<You have choosen ',e2[choice2],' as your Elective3>>>');
                         textcolor(white);
                       end
                else writeln('<<<Duplicate choice>>>')
            else writeln('<<<Duplicate choice>>>')
        else writeln('<<<Not Enough Quota>>>')
     else
          begin choice_check:=true;
                elective2[stud_index]:= e2[choice2];
                writeln;
                textcolor(cyan);
                writeln('<<<You Have Reset Your Elective2>>>');
                textcolor(white);
          end
until choice_check ;
store_students;
store_subject2;
textcolor(white);
writeln;
write('<<<Input <Enter> to Return>>>');
readln;
end;
```

14. choose_e3

This procedure is used to choose the elective1 of the students. It will display the array in the 'block3.txt' first and let the users input their choices. The choice is corresponding to the e3_index. For choice of 0 quotas, same subject as other elective block, the program will ask for input again until users input the valid value. Users can reset their choices by input the number out of the range of e3_index.

```
procedure choose_e3(var stud_index:integer);
var i,j,choice3:integer;
    choice_check:boolean;
begin
clrscr;
j:=find_sub_index3(elective3[stud_index]);
If j<>0 then
            quota_e3[j]:= quota_e3[j]+1;

writeln;
writeln('[ELECTIVE3 SELECTION]              ');
writeln;
writeln('Choice                Elective                Quota    ');
writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><><><> ');
writeln;
i:=1 ;
 for i:= i to num_e3
 do begin
         writeln('',e3_index[i]:3,'',e3[i]:28,'',quota_e3[i]:18);
    end  ;
writeln;
writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><><><> ');
repeat
textcolor(white);
writeln;
writeln('Please Choose Your Elective3') ;
writeln('(1-',num_e3,', Others To Reset Your Choice )');
write('Enter Your Choice: ');
readln(choice3);
writeln;
choice_check :=false;
textcolor(6);

     if (choice3>=1) and (choice3<=num_e3) then
        if ( quota_e3[choice3] >0)          then
            if  (e3[choice3]<>elective1[stud_index]) then
                if  (e3[choice3]<>elective2[stud_index]) then
                        begin
                          choice_check:=true;
                          elective3[stud_index]:= e3[choice3];
                          quota_e3[choice3]:=quota_e3[choice3]-1;
                          textcolor(yellow);
                          writeln('<<<You have choosen ',e3[choice3],' as your Elective3>>>');
                          textcolor(white);
                          end
                else writeln('<<<Duplicate choice>>>')
            else writeln('<<<Duplicate choice>>>')
        else writeln('<<<Not Enough Quota>>>')
     else
           begin choice_check:=true;
                 elective3[stud_index]:= e3[choice3];
                 textcolor(cyan);
                 writeln;
                writeln('<<<You Have Reset Your Elective3>>>');
                textcolor(white);
                end
until choice_check ;
store_students;
store_subject3;
textcolor(white);
writeln;
write('<<<Input <Enter> to Return>>>');
readln;
end;
```

15. find_sub_index1 find_sub_index2 find_sub_index3

When users are choosing an elective, the original choice of the users should be reset and add 1 quota back to the corresponding subject. This function is find out the place of the array of the original subject and return the value as an indicator to add back 1 quota. These functions are used in choose_e1, choose_e2 and choose_e3 respectively

```
function find_sub_index1(subname:string):integer;
var i:integer;
begin
 find_sub_index1:=0;
 for i:= 1 to num_e1 do
    if e1[i]=subname then
       find_sub_index1:=i
end;

function find_sub_index2(subname:string):integer;
var i:integer;
begin
 find_sub_index2:=0;
 for i:= 1 to num_e2 do
    if e2[i]=subname then
       find_sub_index2:=i
end;

function find_sub_index3(subname:string):integer;
var i:integer;
begin
 find_sub_index3:=0;
 for i:= 1 to num_e3 do
    if e3[i]=subname then
       find_sub_index3:=i
end;
```

3.4 Program Coding

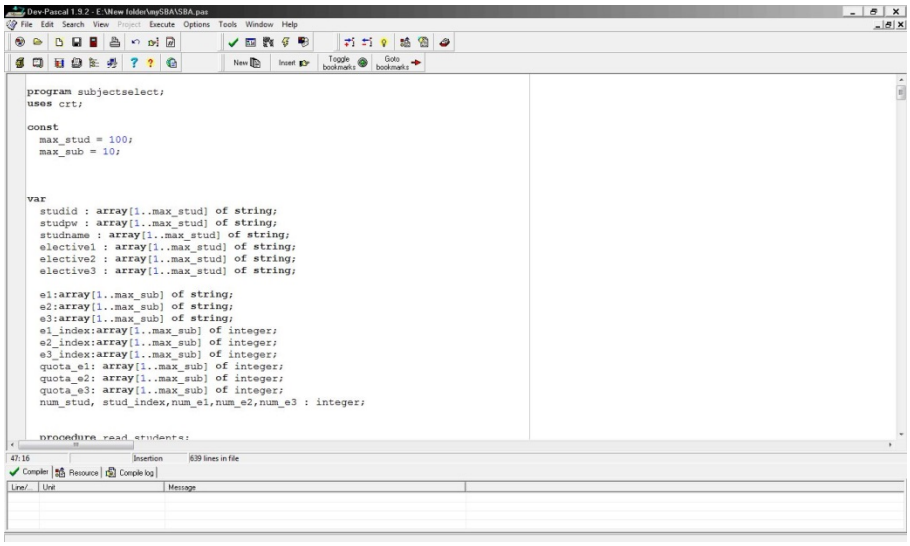
My program uses pascal language for the program coding and the tool used is Dev-pascal. The program consists of main body, functions and procedures.

The name of the source program is SBA.pas.

The name of the object program is SBA.exe.

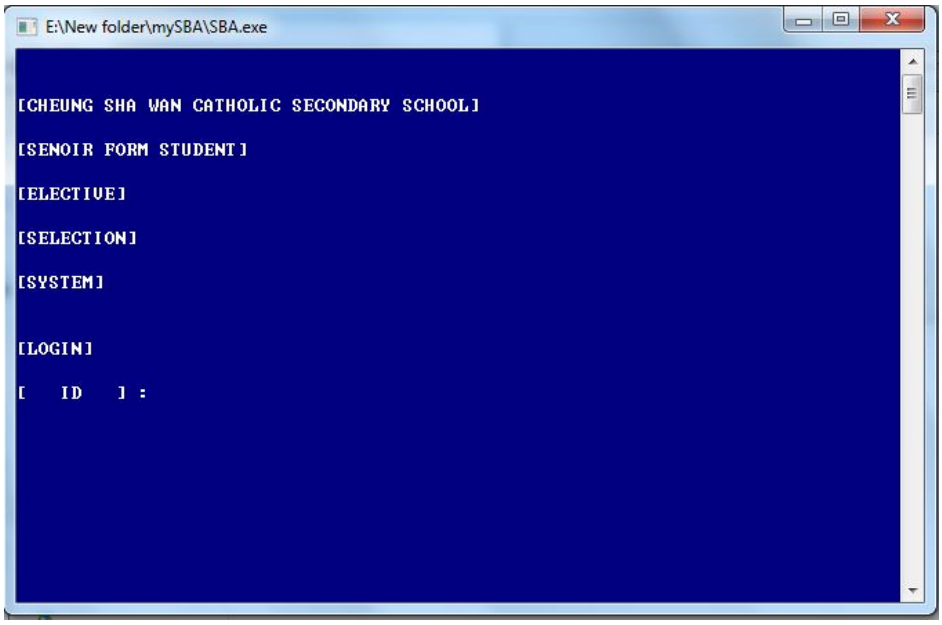
The program requires 4 text files to carry out normal execution including studentslist.txt, block1.txt, block2.txt, block3.txt.

Here is the sample coding of my program (full coding refer to appendix)



```
program subjectselect;
uses crt;

const
  max_stud = 100;
  max_sub = 10;


var
  studid : array[1..max_stud] of string;
  studpw : array[1..max_stud] of string;
  studname : array[1..max_stud] of string;
  elective1 : array[1..max_stud] of string;
  elective2 : array[1..max_stud] of string;
  elective3 : array[1..max_stud] of string;

  e1:array[1..max_sub] of string;
  e2:array[1..max_sub] of string;
  e3:array[1..max_sub] of string;
  e1_index:array[1..max_sub] of integer;
  e2_index:array[1..max_sub] of integer;
  e3_index:array[1..max_sub] of integer;
  quota_e1: array[1..max_sub] of integer;
  quota_e2: array[1..max_sub] of integer;
  quota_e3: array[1..max_sub] of integer;
  num_stud, stud_index,num_e1,num_e2,num_e3 : integer;

  procedure read_students;
```

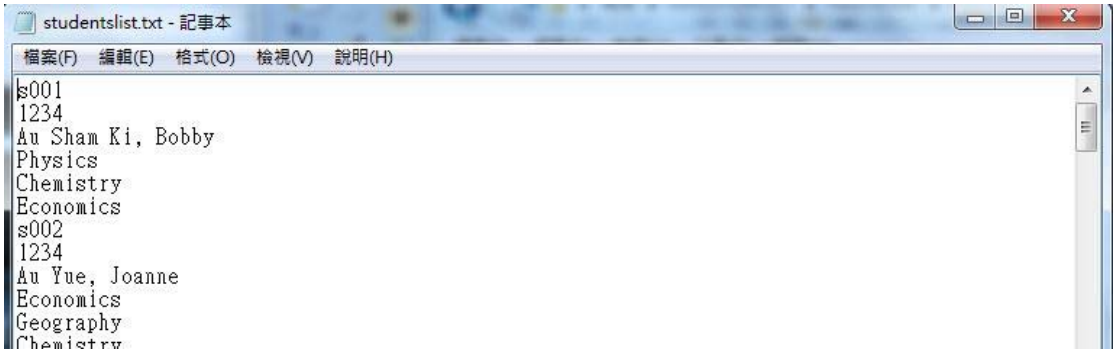
Here is the sample execution of my program

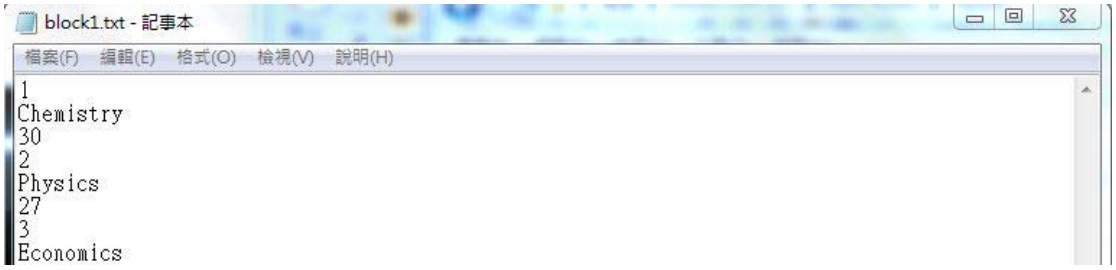


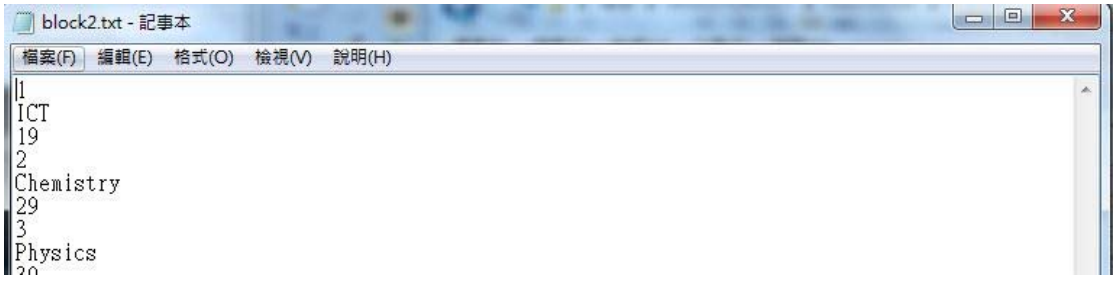
```
E:\New folder\mySBA\SBA.exe

[CHEUNG SHA WAN CATHOLIC SECONDARY SCHOOL]

[SENOIR FORM STUDENT]

[ELECTIVE]

[SELECTION]

[SYSTEM]

[LOGIN]

[   ID   ] :
```

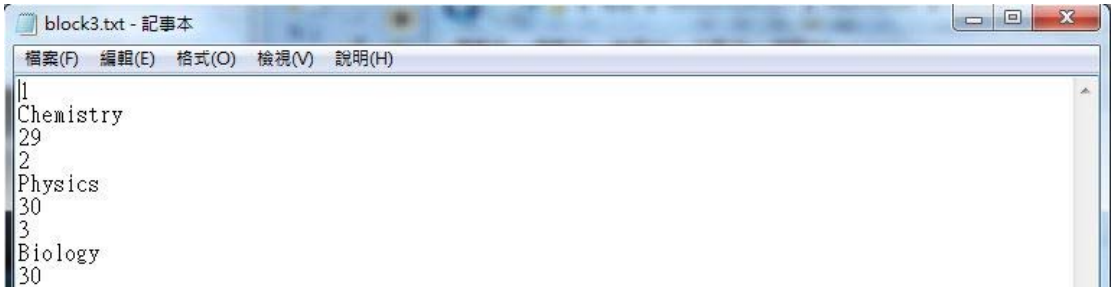
3.5 Program execution

1. Program file: SBA.EXE
2. Data file to be prepare for input:
   - The text file storing the students' data: studentslist.txt
   - The text file storing the elective1 data: block1.txt
   - The text file storing the elective2 data: block2.txt
   - The text file storing the elective3 data: block3.txt
   - The program file and data files should be put into the same folder / location before execution.

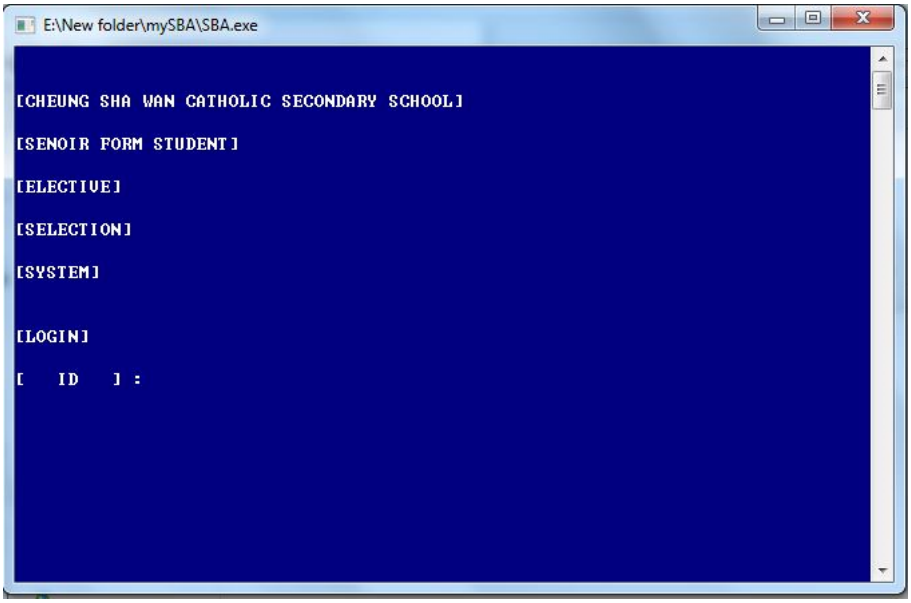






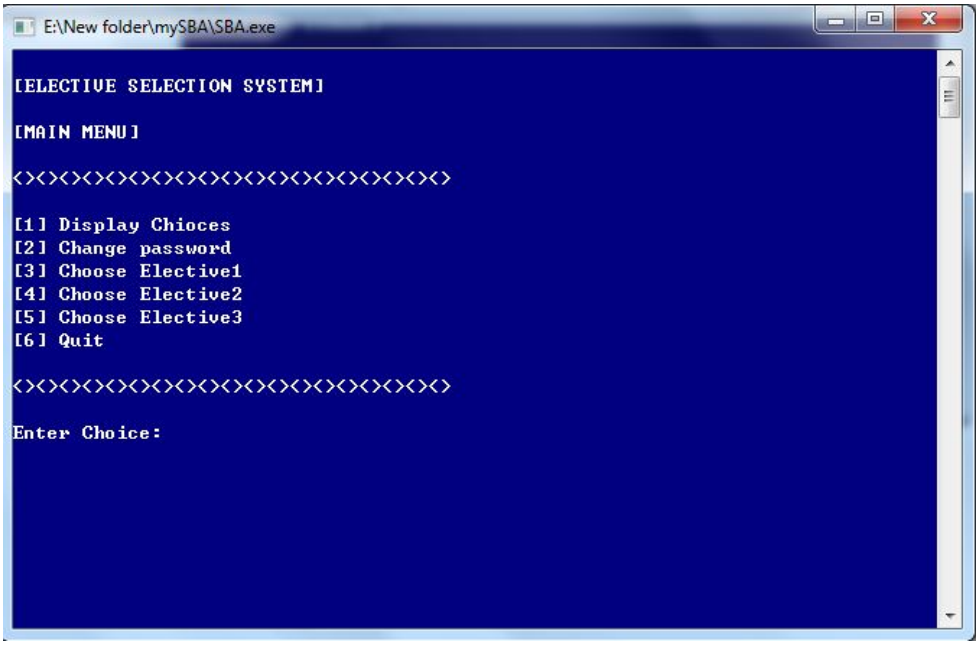


User interface for login



Users need to login to the system by using their own student id and passwords.
After input data, users need to press enter to proceed to next stage.
Wrong ID or password will return 'Invalid Input' and ask for input again.

Main menu

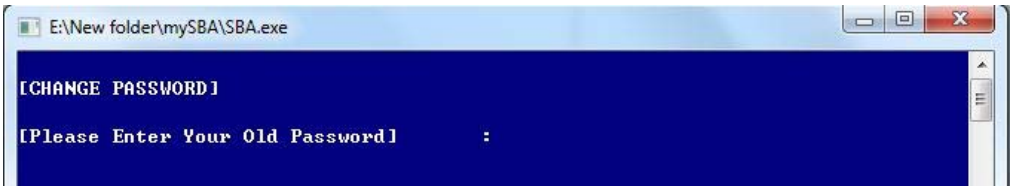


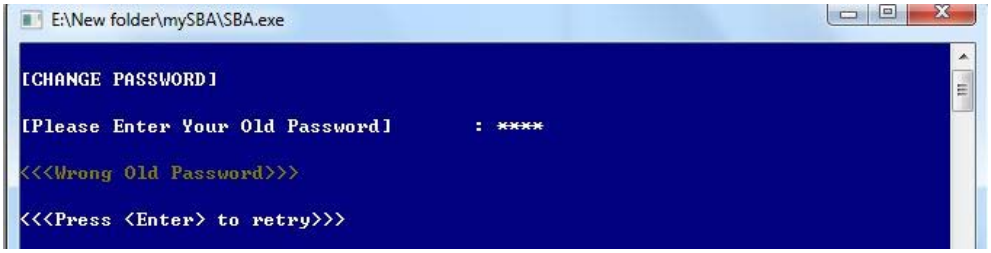
After successfully login to the system, users need to input integer according to the options display on the main menu.

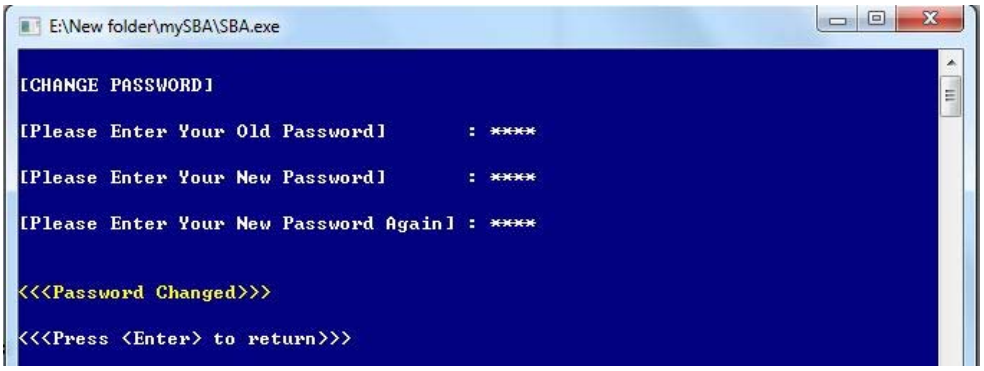
Display Choice

This interface will display the data which is stored in studentslist.txt. Press enter will return to main menu.

Change Password

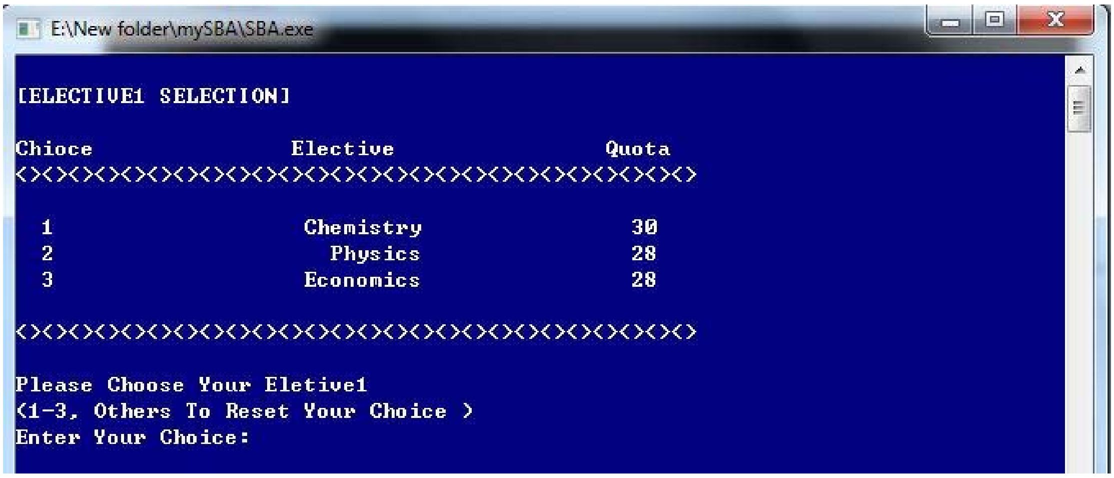




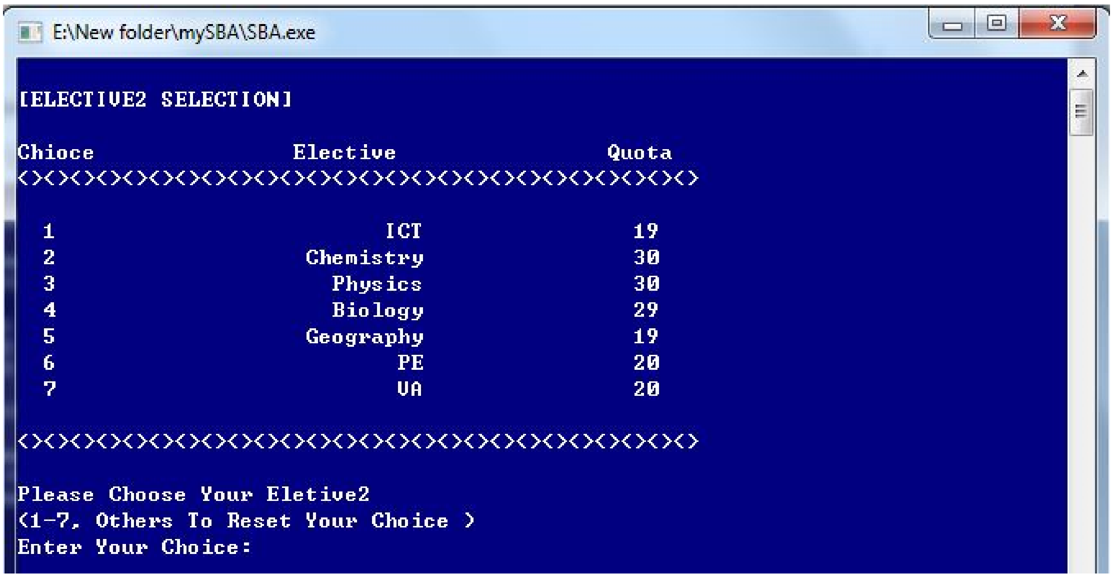


Users can change their passwords in this procedure. The screenshots show different cases of input.

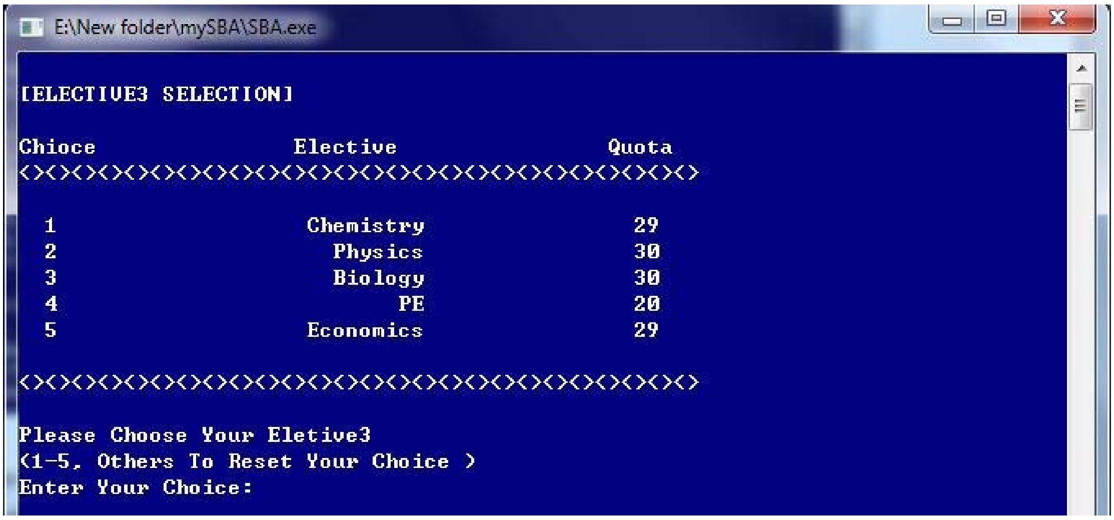
Choose Elective1 & Choose Elective2 & Choose Elective3



```
[ELECTIVE1 SELECTION]

Chioce               Elective                Quota
<><><><><><><><><><><><><><><><><><><><><><><><><><>

  1                   Chemistry               30
  2                     Physics               28
  3                   Economics               28

<><><><><><><><><><><><><><><><><><><><><><><><><><>

Please Choose Your Eletive1
(1-3, Others To Reset Your Choice )
Enter Your Choice:
```



```
[ELECTIVE2 SELECTION]

Chioce               Elective                Quota
<><><><><><><><><><><><><><><><><><><><><><><><><><>

  1                         ICT               19
  2                   Chemistry               30
  3                     Physics               30
  4                     Biology               29
  5                   Geography               19
  6                          PE               20
  7                          VA               20

<><><><><><><><><><><><><><><><><><><><><><><><><><>

Please Choose Your Eletive2
(1-7, Others To Reset Your Choice )
Enter Your Choice:
```



```
[ELECTIVE3 SELECTION]

Chioce               Elective                Quota
<><><><><><><><><><><><><><><><><><><><><><><><><><>

  1                   Chemistry               29
  2                     Physics               30
  3                     Biology               30
  4                          PE               20
  5                   Economics               29

<><><><><><><><><><><><><><><><><><><><><><><><><><>

Please Choose Your Eletive3
(1-5, Others To Reset Your Choice )
Enter Your Choice:
```

Users need to input the corresponding numeric value of the subject for choices. Duplicate choices or zero-quota will return error and ask for new input again.

Inputting value out of the range of the choices will reset the elective selection and release the quota.

Chapter 4 Testing & Evaluation

4.1 Brief Description

In this chapter, I will discuss the testing and evaluation of the program. The purpose is to find out the bugs in the program including logical errors and runtime errors and also to check whether the program achieve it purpose of subject selection. Moreover, I will find out if the program is user-friendly or not and make further improvement about it.

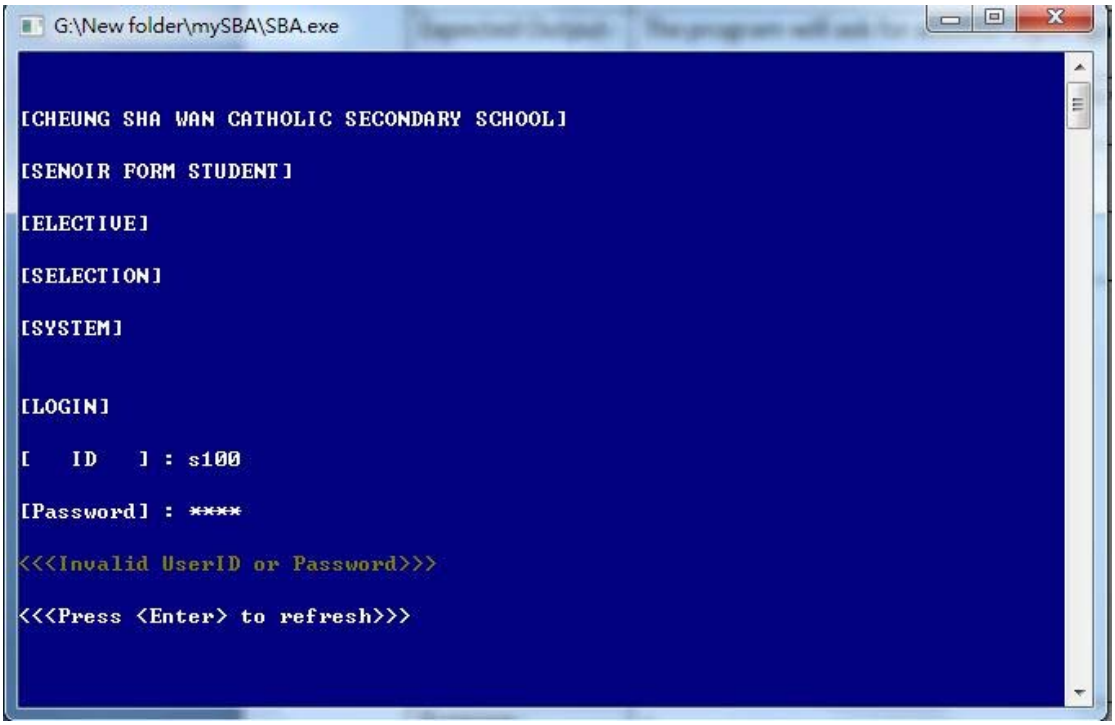
4.2 Testing and Evaluation Plan

The program will be tested according to the following plan:

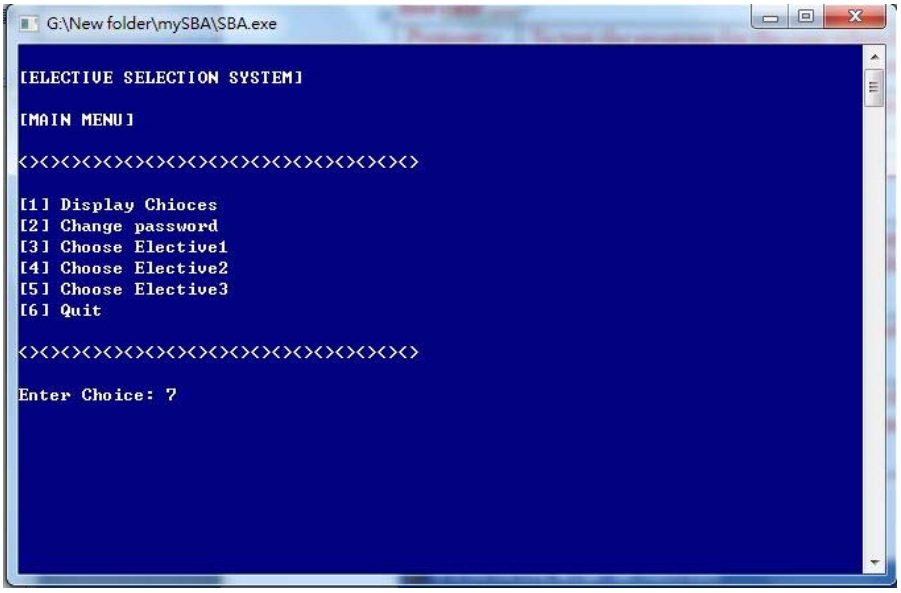
1. Internal Testing
   - The internal testing will be carried out by the programmers(by me)
   - I will prepare different cases of input to test the program if there are bugs
   - The test cases include some correct input data (from files & keyboard) with known results for checking the correctness of the program, some incorrect input data to see whether the program can handle invalid input reasonably.
   - I will also evaluate the programs according to its user-friendless, performance, flexibility for future development, reusability of program codes
2. External Testing
   - I will invite some targeted users (my classmates/friends) to test and evaluate the program.
   - I will upload the object program to E-class platform for the external tester to test
   - I will upload the program to different social networking sites and forums.
   - The testers will report bugs and feedbacks after testing.
   - I will try to improve and modify my program according to their opinions.

4.3 Internal Testing

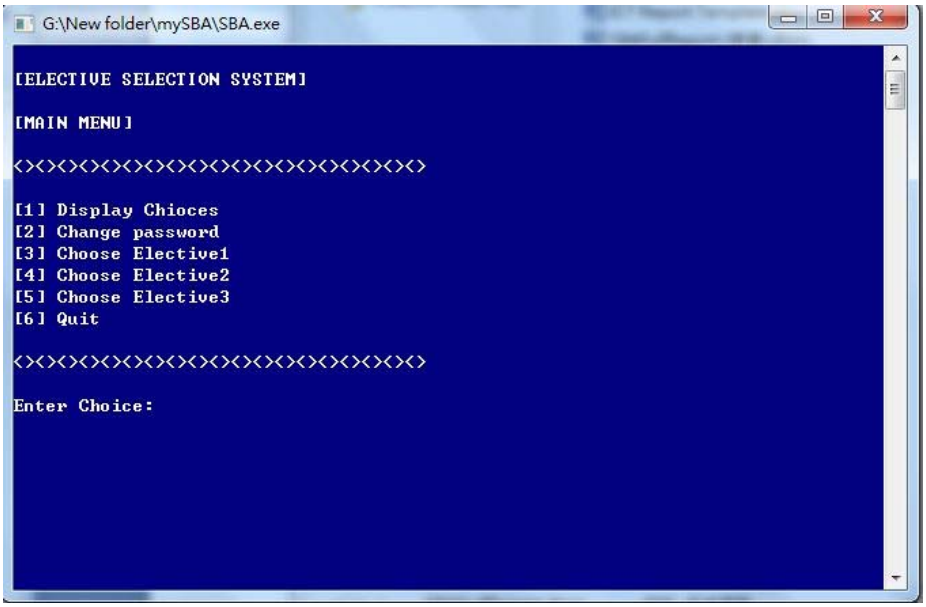
Purpose	Test the program when inputting incorrect login is or password
Input	Fake login id and password
Expected output	The program will show error and ask to input ask
Actual output	The program asks for input again as shown below
Test result	No bugs found
Follow-up action	Nil



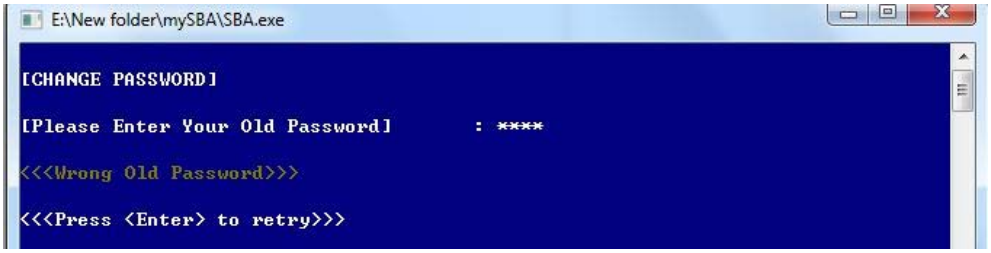
Purpose	To test the program when inputting the value out of the range
Input	Enter ‘7’ in the main menu
Expected output	The program will ask for valid input again
Actual output	The program refreshes when inputting ‘7’, users can input the choices again
Test result	No bugs found
Follow-up action	Nil

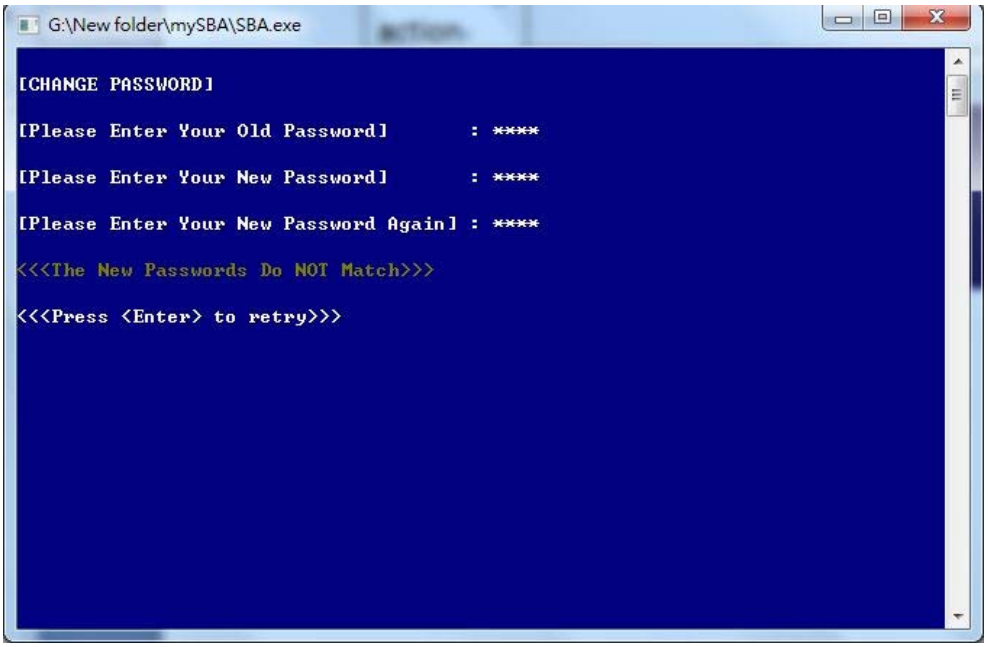


After press enter

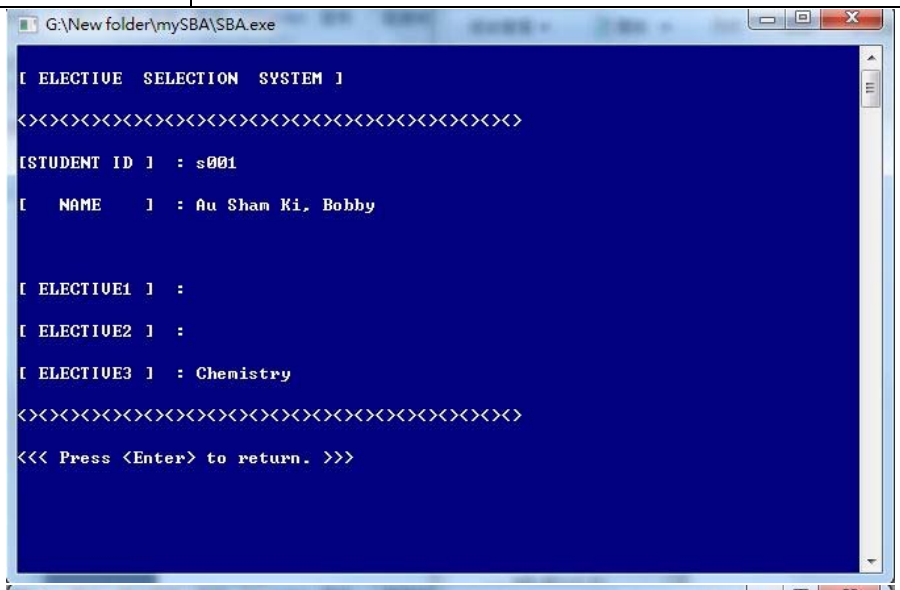


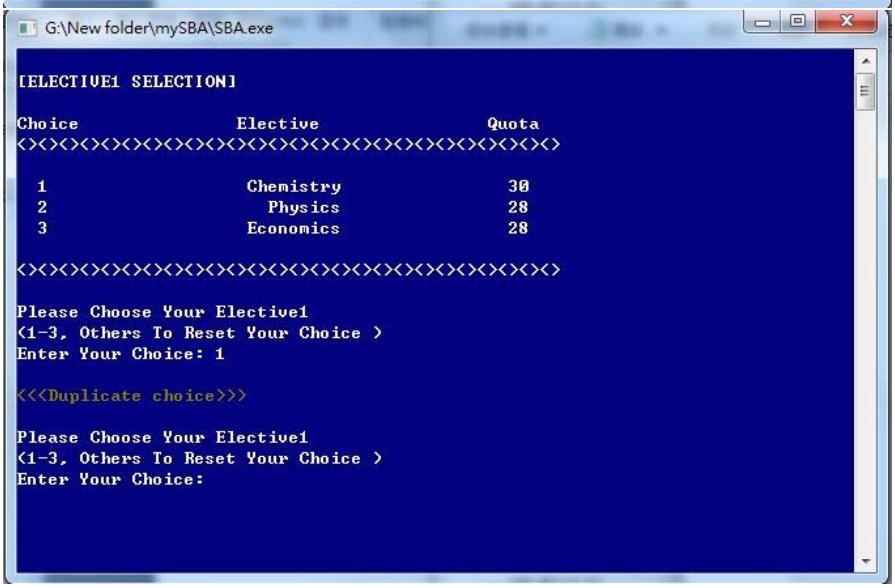
Purpose	To test the program when inputting incorrect password in password changing
Input	Wrong old password and two different new passwords
Expected output	The program will return errors and ask for input again
Actual output	The program returns errors and ask for input again
Test result	No bugs found
Follow-up action	Nil



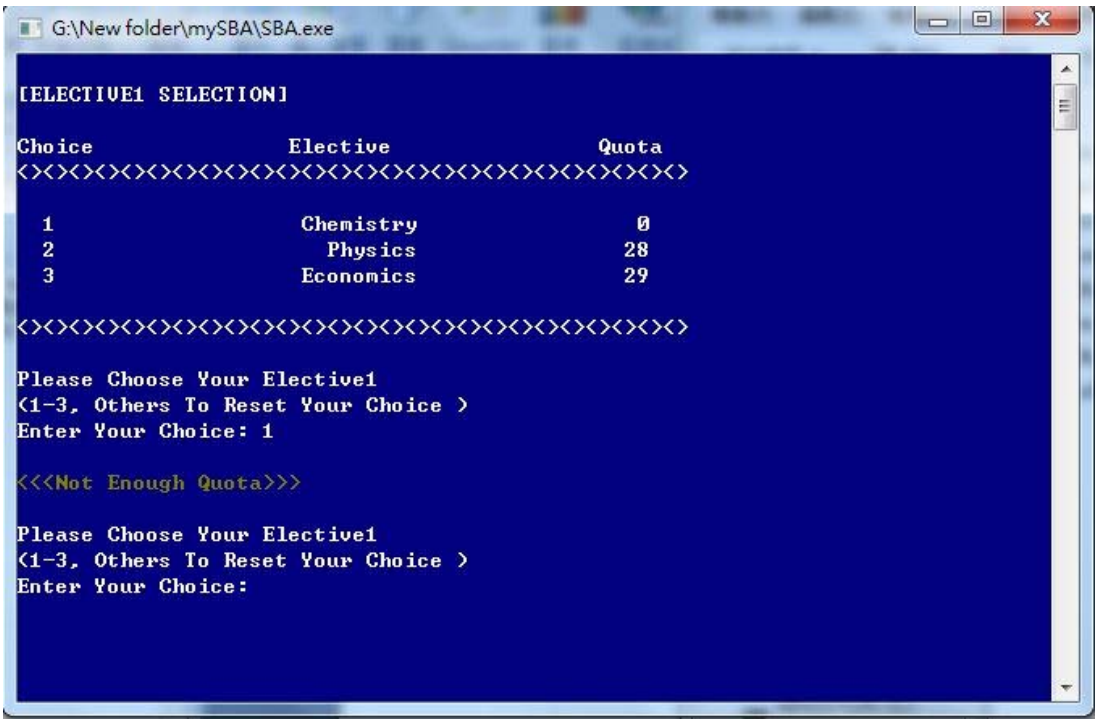


Purpose	To test the program when the users choose duplicate subject in different blocks of electives
Input	Choose chemistry in elective 1 (initially he has chosen chemistry for elective 3)
Expected output	The program will show duplicate choice and ask for input again
Actual output	The program shows duplicate choice and ask for input again
Test result	No bugs found
Follow-up action	Nil



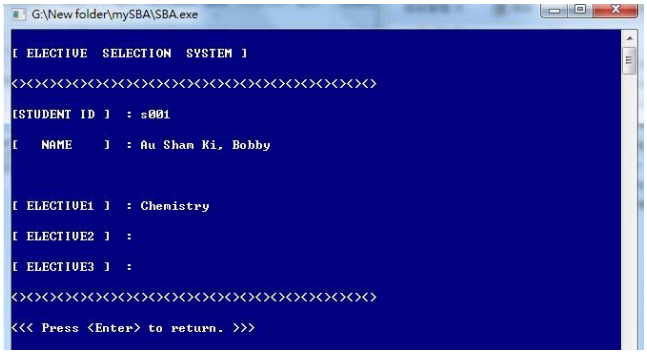


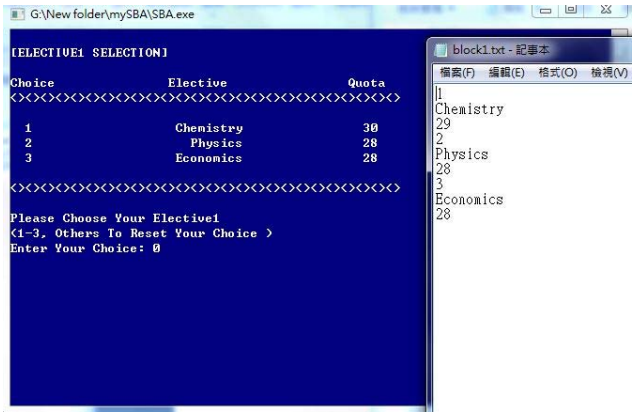
Purpose	To test the program when the users choose the subject which has no quotas
Input	Choose chemistry by input 1
Expected output	The program will show not enough quotas and ask for input again
Actual output	The program shows not enough quotas and ask for input again
Test result	No bugs found
Follow-up action	Nil

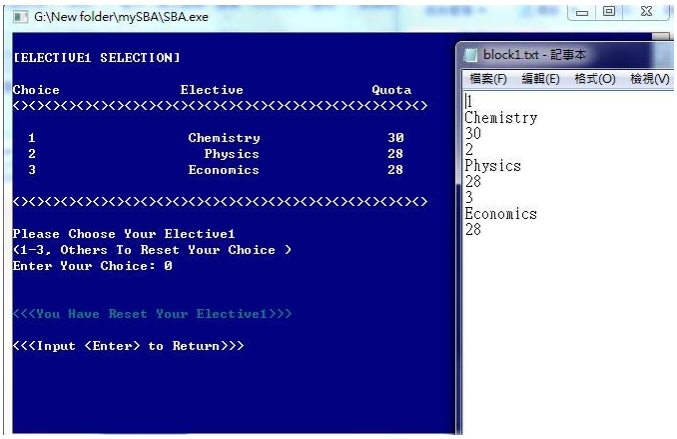


```
[ELECTIVE1 SELECTION]

Choice               Elective                 Quota
<><><><><><><><><><><><><><><><><><><><><><><><><><>

  1                   Chemistry                  0
  2                     Physics                 28
  3                   Economics                 29

<><><><><><><><><><><><><><><><><><><><><><><><><><>

Please Choose Your Elective1
(1-3, Others To Reset Your Choice )
Enter Your Choice: 1

<<<Not Enough Quota>>>

Please Choose Your Elective1
(1-3, Others To Reset Your Choice )
Enter Your Choice:
```

Purpose	To test the program when the users input the value of the range
Input	Input 4 in elective 1
Expected output	The program will reset the choice of elective 1, the quota of the corresponding will add 1
Actual output	The program shows ‘ you have reset your elective 1’, the quota of the corresponding adds 1.
Test result	No bugs found
Follow-up action	Nil







Purpose	To test the program when the users input the value of other type of values
Input	Input character when it asks for numeric input
Expected output	The program will not run anything
Actual output	The program is force closed
Test result	Bugs found
Follow-up action	Need further modifications

4.4 Self-Evaluation

My program will run normally when the users enter the numeric integer value in different stages but it will force closed when inputting different type of the input values.

My program is use friendly as the users can choose to select their elective in different parts. They need not to choose all electives at one time. When they want to choose elective 2, they need not to choose elective 1 first. So it brings convenience to users.

The reusability of my program is also high. When the user wants to choose elective again, my program will restore the quotas of the subject being chosen by the user before. Thus, the users can choose their elective for more than 1 time.

4.5 External Testing and Evaluation

To perform external testing, I have uploaded my program packages (.exe files and text files) to E-class platform for other classmates to test. The classmates have knowledge about programming so that they can test my program comprehensively and give instant feedback of my program to me.

Chapter 5 Conclusion & Discussion

5.1 Pros and Cons of my program

For the pros of my program, it provides a convenient and efficient way for users to choose their elective. It needs not human recording of the data. It also provides higher security of the personal data as my program requires the login id and password of the user when belongs to certain person only. Moreover, my program can prevent duplicate choices and disallow users choose the subjects which have 0 quotas. The data will be tidied up and save in a text file according to student number in ascending order.

For the cons of program, the text files must be placed in the location of the program so that users can easily edit the value of the text files. Also, my program will force closed in case of undefined input values.

5.2 Further Improvement

I will try to tackle the problem of force closed in case of undefined input values. And also adding an administrator account or system to perform maintenance of the program and check the record of the students. Moreover, I will try to raise the security of the text files as to prevent editions by the users.

5.3 Reflection

In writing this subject selection program, I have learnt that the combination of different procedures and logical sequences. And it is not easy to write a program which is free of bugs, the program codes are complicated and it is very difficult to check out the bugs especially the logical errors and run-time errors. I have learnt a lot of debug methods in this program writing task.

Chapter 6 Reference and Acknowledgement

From Internet websites

1. http://computer-programming-forum.com/29-pascal/7af4f3f05f738777.htm

Acknowledgement

1. My teacher supervisor Mr. Chu

Appendices

Appendix 1 program codes

```
program subjectselect;
uses crt;

const
   max_stud = 100;
   max_sub = 10;



var
   studid : array[1..max_stud] of string;
   studpw : array[1..max_stud] of string;
   studname : array[1..max_stud] of string;
   elective1 : array[1..max_stud] of string;
   elective2 : array[1..max_stud] of string;
   elective3 : array[1..max_stud] of string;

   e1:array[1..max_sub] of string;
   e2:array[1..max_sub] of string;
```

```
  e3:array[1..max_sub] of string;
  e1_index:array[1..max_sub] of integer;
  e2_index:array[1..max_sub] of integer;
  e3_index:array[1..max_sub] of integer;
  quota_e1: array[1..max_sub] of integer;
  quota_e2: array[1..max_sub] of integer;
  quota_e3: array[1..max_sub] of integer;
  num_stud, stud_index,num_e1,num_e2,num_e3 : integer;



  procedure read_students;
var
  i : integer;
  studlist : text;
begin
  assign(studlist, 'studentslist.txt');
  reset(studlist);
  i := 0;
  while not eof(studlist) do
    begin
      i := i + 1;
      readln(studlist, studid[i]);
      readln(studlist, studpw[i]);
```

```
        readln(studlist, studname[i]);
        readln(studlist, elective1[i]);
        readln(studlist, elective2[i]);
        readln(studlist, elective3[i]);
      end;
   num_stud := i;
   close(studlist)
end;


 procedure   store_subject1;
var   i : integer;
       block1 : text;
begin
   assign(block1, 'block1.txt');
   rewrite(block1);
   for i := 1 to num_e1 do
      begin
        writeln(block1, e1_index[i]);
        writeln(block1, e1[i]);
        writeln(block1, quota_e1[i]);
      end;
   close(block1)
end;
```

```
   procedure   store_subject2;
var   i : integer;
        block2 : text;
begin
   assign(block2, 'block2.txt');
   rewrite(block2);
   for i := 1 to num_e2 do
      begin
         writeln(block2, e2_index[i]);
         writeln(block2, e2[i]);
         writeln(block2, quota_e2[i]);
      end;
   close(block2)
end;

  procedure   store_subject3;
var   i : integer;
        block3 : text;
begin
   assign(block3, 'block3.txt');
   rewrite(block3);
   for i := 1 to num_e3 do
```

```
      begin
        writeln(block3, e3_index[i]);
        writeln(block3, e3[i]);
        writeln(block3, quota_e3[i]);
      end;
   close(block3)
end;


   procedure read_e1;
var
   i : integer;
   block1 : text;
begin
   assign(block1, 'block1.txt');
   reset(block1);
   i := 0;
   while not eof(block1) do
      begin
        i := i + 1;
        readln(block1, e1_index[i]);
        readln(block1, e1[i]);
        readln(block1, quota_e1[i]);
```

```
      end;
   num_e1 := i;
   close(block1)
end;


    procedure read_e2;
var
   i : integer;
   block2 : text;
begin
   assign(block2, 'block2.txt');
   reset(block2);
   i := 0;
   while not eof(block2) do
      begin
         i := i + 1;
         readln(block2, e2_index[i]);
         readln(block2, e2[i]);
         readln(block2, quota_e2[i]);
```

```
      end;
   num_e2 := i;
   close(block2)
end;




 procedure read_e3;
var
   i : integer;
   block3 : text;
begin
   assign(block3, 'block3.txt');
   reset(block3);
   i := 0;
   while not eof(block3) do
      begin
         i := i + 1;
         readln(block3, e3_index[i]);
         readln(block3, e3[i]);
         readln(block3, quota_e3[i]);

      end;
```

```
   num_e3 := i;
   close(block3)
end;




     procedure store_students;
var   i : integer;
        studlist : text;
begin
   assign(studlist, 'studentslist.txt');
   rewrite(studlist);
   for i := 1 to num_stud do
      begin
         writeln(studlist, studid[i]);
         writeln(studlist, studpw[i]);
         writeln(studlist, studname[i]);
         writeln(studlist, elective1[i]);
         writeln(studlist, elective2[i]);
         writeln(studlist, elective3[i]);
      end;
   close(studlist)
end;
```

```
{ Ref:
http://computer-programming-forum.com/29-pascal/7af4f3f
05f738777.htm }

function GetPWord : string;     (* A function for hiding
password *)

var

   S : string;

   C : Char;

begin

   S := '';

   repeat

      C := ReadKey;

      if (C <> #10) and (C <> #13) and (C <> #8) then

         begin

            S := S + C;

            write('*');

         end

      else if C = #8 then
```

```
        begin
          S[0] := Chr(Length(S) - 1);
          GotoXY(WhereX - 1, WhereY);
          write(' ');
          GotoXY(WhereX - 1, WhereY);
        end;
  until (C = #10) or (C = #13);
  GetPWord := S;
  writeLn;
end;




procedure login(var stud_index : integer);
var
  userid, password : string;
  found : boolean;
  i : integer;
begin
  clrscr;
  writeln;
  writeln;
```

```
    textcolor(white);
    writeln('[CHEUNG SHA WAN CATHOLIC SECONDARY
SCHOOL]');
    writeln('');
    writeln('[SENOIR FORM STUDENT]');
    writeln('');
    writeln('[ELECTIVE]');
    writeln('');
    writeln('[SELECTION]');
    writeln('');
    writeln('[SYSTEM]');
    textcolor(white);
    writeln;
    writeln;
    writeln('[LOGIN]');
    writeln;
    write('[     ID     ] : ');
    readln(userid);
    writeln;
    write('[Password] : ');
    password := GetPword;
    writeln;
    found := false;
    i := 0;
```

```
  while (i < num_stud) and (not found) do
    begin
      i := i + 1;
      if (userid = studid[i]) and (password = studpw[i]) then
        begin
          found := true;
          stud_index := i
        end
    end;
  if not found then
    begin
      stud_index := 0;
      textcolor(6);
      writeln('<<<Invalid UserID or Password>>>');
      writeln;
      textcolor(white);
      write('<<<Press <Enter> to refresh>>>');
      readln
    end;
end;
```

```
procedure display(stud_index : integer);
begin
   clrscr;
   writeln;
   textcolor(white);
   writeln('[ ELECTIVE   SELECTION   SYSTEM ]');
   writeln;

writeln('<><><><><><><><><><><><><><><><><><><><><><><><><><><><>
<><><><> ');
   writeln;
   writeln('[STUDENT ID ]   : ', studid[stud_index]);
   writeln;
   writeln('[     NAME      ]   : ', studname[stud_index]);
   writeln;
   writeln;
   writeln;
   writeln('[ ELECTIVE1 ]   : ', elective1[stud_index]);
   writeln;
   writeln('[ ELECTIVE2 ]   : ', elective2[stud_index]);
   writeln;
   writeln('[ ELECTIVE3 ]   : ', elective3[stud_index]);
   writeln;
```

```
writeln('<><><><><><><><><><><><><><><><><><><><><>
<><><> ');
   writeln;
   textcolor(white);
   write('<<< Press <Enter> to return. >>>');
   readln ;

end;


procedure change_password(stud_index : integer);
var
   oldpass, newpass1, newpass2 : string;
   pwchanged : boolean;
begin
   pwchanged := false;
   repeat
      clrscr;
```

```
writeln;
writeln('[CHANGE PASSWORD]');
WRITELN;
write('[Please Enter Your Old Password]          : ');
oldpass := GetPword;
if oldpass <> studpw[stud_index] then
   begin
      writeln;
      TEXTCOLOR(6);
      writeln('<<<Wrong Old Password>>>');
      TEXTCOLOR(WHITE);
      writeln;
      write('<<<Press <Enter> to retry>>>');
      readln
   end
else
   begin
      writeln;
      write('[Please Enter Your New Password]          : ');
      newpass1 := GetPword;
      writeln;
      write('[Please Enter Your New Password Again] : ');
      newpass2 := GetPword;
```

```
            if newpass1 <> newpass2 then
              begin
                writeln;
                TEXTCOLOR(6);
                writeln('<<<The New Passwords Do NOT
Match>>>');
                TEXTCOLOR(WHITE);
                writeln;
                write('<<<Press <Enter> to retry>>>');
                readln
              end
            else
              begin
                studpw[stud_index] := newpass1;
                store_students;
                pwchanged := true;
                writeln;
                TEXTCOLOR(yellow);
                writeln;
                writeln('<<<Password Changed>>>');
                TEXTCOLOR(WHITE);
                writeln;
                write('<<<Press <Enter> to return>>>');
                readln
```

```
            end
        end

  until pwchanged
end;



function find_sub_index1(subname:string):integer;
var i:integer;
begin
 find_sub_index1:=0;
 for i:= 1 to num_e1 do
     if e1[i]=subname then
         find_sub_index1:=i
end;

function find_sub_index2(subname:string):integer;
var i:integer;
begin
 find_sub_index2:=0;
 for i:= 1 to num_e2 do
     if e2[i]=subname then
         find_sub_index2:=i
```

```
end;

function find_sub_index3(subname:string):integer;
var i:integer;
begin
 find_sub_index3:=0;
 for i:= 1 to num_e3 do
     if e3[i]=subname then
         find_sub_index3:=i
end;


 procedure choose_e1(var stud_index:integer);
 var i,j,choice1:integer;
       choice_check:boolean;
 begin
 clrscr;
 j:=find_sub_index1(elective1[stud_index]);
 If j<>0 then
                    quota_e1[j]:= quota_e1[j]+1;

 writeln;
 writeln('[ELECTIVE1 SELECTION]                    ');
```

```
  writeln;

  writeln('Choice                    Elective
Quota       ');


writeln('<><><><><><><><><><><><><><><><><><><><><><><><><>
<><><><><><> ');

  writeln;

  i:=1 ;

   for i:= i to num_e1

   do begin

              writeln(',e1_index[i]:3,',e1[i]:28,',quota_e1[i]:18);

       end   ;

  writeln;


writeln('<><><><><><><><><><><><><><><><><><><><><><><><><>
<><><><><><> ');

  repeat

  textcolor(white);

  writeln;

  writeln('Please Choose Your Elective1') ;

  writeln('(1-',num_e1,', Others To Reset Your Choice )');

  write('Enter Your Choice: ');

  readln(choice1);

  writeln;

  choice_check :=false;
```

```
 textcolor(6);
        if (choice1>=1) and (choice1<=num_e1) then
            if ( quota_e1[choice1] >0)              then
                if   (e1[choice1]<>elective2[stud_index]) then
                     if   (e1[choice1]<>elective3[stud_index])
then
                              begin
                                 choice_check:=true;
                                 elective1[stud_index]:=
e1[choice1];

quota_e1[choice1]:=quota_e1[choice1]-1;
                                 textcolor(yellow);
                                 writeln('<<<You have choosen
',e1[choice1],' as your Elective1>>>');
                                 textcolor(white);
                              end
                     else writeln('<<<Duplicate choice>>>')
                else writeln('<<<Duplicate choice>>>')
            else writeln('<<<Not Enough Quota>>>')
        else
                begin choice_check:=true;
                      elective1[stud_index]:= e1[choice1];
                      textcolor(cyan);
                      writeln;
```

```
                    writeln('<<<You Have Reset Your
Elective1>>>');
                    textcolor(white);
              end
 until choice_check ;
 store_students;
 store_subject1;
 textcolor(white);
 writeln;
 write('<<<Input <Enter> to Return>>>');
 readln;
 end;

 procedure choose_e2(var stud_index:integer);
 var i,j,choice2:integer;
     choice_check:boolean;
 begin
 clrscr;
 j:=find_sub_index2(elective2[stud_index]);
 If j<>0 then
              quota_e2[j]:= quota_e2[j]+1;

 writeln;
 writeln('[ELECTIVE2 SELECTION]                    ');
```

```
  writeln;

  writeln('Choice                    Elective
Quota       ');


writeln('<><><><><><><><><><><><><><><><><><><><><><><><><>
<><><><><><> ');

  writeln;

  i:=1 ;

   for i:= i to num_e2

   do begin

              writeln(',e2_index[i]:3,',e2[i]:28,',quota_e2[i]:18);

        end    ;

  writeln;


writeln('<><><><><><><><><><><><><><><><><><><><><><><><><>
<><><><><><> ');

  repeat

  textcolor(white);

  writeln;

  writeln('Please Choose Your Elective2') ;

  writeln('(1-',num_e2,', Others To Reset Your Choice )');

  write('Enter Your Choice: ');

  readln(choice2);

  writeln;

  choice_check :=false;
```

```
 textcolor(6);
        if (choice2>=1) and (choice2<=num_e2) then
            if ( quota_e2[choice2] >0)              then
                if   (e2[choice2]<>elective1[stud_index]) then
                    if   (e2[choice2]<>elective3[stud_index])
then
                            begin
                              choice_check:=true;
                              elective2[stud_index]:=
e2[choice2];

quota_e2[choice2]:=quota_e2[choice2]-1;
                              textcolor(yellow);
                              writeln('<<<You have choosen
',e2[choice2],' as your Elective3>>>');
                              textcolor(white);
                            end
                    else writeln('<<<Duplicate choice>>>')
                else writeln('<<<Duplicate choice>>>')
            else writeln('<<<Not Enough Quota>>>')
        else
                begin choice_check:=true;
                    elective2[stud_index]:= e2[choice2];
                    writeln;
                    textcolor(cyan);
```

```
                    writeln('<<<You Have Reset Your
Elective2>>>');
                    textcolor(white);
            end
 until choice_check ;
 store_students;
 store_subject2;
 textcolor(white);
 writeln;
 write('<<<Input <Enter> to Return>>>');
 readln;
 end;

 procedure choose_e3(var stud_index:integer);
 var i,j,choice3:integer;
     choice_check:boolean;
 begin
 clrscr;
 j:=find_sub_index3(elective3[stud_index]);
 If j<>0 then
            quota_e3[j]:= quota_e3[j]+1;

 writeln;
 writeln('[ELECTIVE3 SELECTION]                    ');
```

```
  writeln;

  writeln('Choice                    Elective
Quota       ');


writeln('<><><><><><><><><><><><><><><><><><><><><><><><><>
<><><><><><> ');

  writeln;

  i:=1 ;

   for i:= i to num_e3

   do begin

              writeln(',e3_index[i]:3,',e3[i]:28,',quota_e3[i]:18);

        end    ;

  writeln;


writeln('<><><><><><><><><><><><><><><><><><><><><><><><><>
<><><><><><> ');

  repeat

  textcolor(white);

  writeln;

  writeln('Please Choose Your Elective3') ;

  writeln('(1-',num_e3,', Others To Reset Your Choice )');

  write('Enter Your Choice: ');

  readln(choice3);

  writeln;

  choice_check :=false;
```

```
 textcolor(6);
        if (choice3>=1) and (choice3<=num_e3) then
            if ( quota_e3[choice3] >0)              then
                if   (e3[choice3]<>elective1[stud_index]) then
                     if   (e3[choice3]<>elective2[stud_index])
then
                              begin
                                 choice_check:=true;
                                 elective3[stud_index]:=
e3[choice3];

quota_e3[choice3]:=quota_e3[choice3]-1;
                                 textcolor(yellow);
                                 writeln('<<<You have choosen
',e3[choice3],' as your Elective3>>>');
                                 textcolor(white);
                                 end
                     else writeln('<<<Duplicate choice>>>')
                else writeln('<<<Duplicate choice>>>')
            else writeln('<<<Not Enough Quota>>>')
        else
                begin choice_check:=true;
                      elective3[stud_index]:= e3[choice3];
                      textcolor(cyan);
                      writeln;
```

```
                        writeln('<<<You Have Reset Your
Elective3>>>');
                        textcolor(white);
                        end
  until choice_check ;
  store_students;
  store_subject3;
  textcolor(white);
  writeln;
  write('<<<Input <Enter> to Return>>>');
  readln;
  end;

procedure main_menu(stud_index : integer);
var
   choice : integer;
begin
   repeat
      clrscr;
      writeln;
      writeln('[ELECTIVE SELECTION SYSTEM]');
      writeln;
      writeln('[MAIN MENU]');
      writeln;
```

```
    writeln('<><><><><><><><><><><><><><><><><><><><><><>');
    writeln;
    writeln('[1] Display Chioces');
    writeln('[2] Change password');
    writeln('[3] Choose Elective1');
    writeln('[4] Choose Elective2');
    writeln('[5] Choose Elective3');
    writeln('[6] Quit');
    writeln;
    writeln('<><><><><><><><><><><><><><><><><><><><><><>');
    writeln;
    write('Enter Choice: ');
    readln(Choice);
   case choice of
      1 : display(stud_index);
      2 : change_password(stud_index);
      3 : choose_e1(stud_index);
      4 : choose_e2(stud_index);
      5 : choose_e3(stud_index);
    end;
    until(choice = 6);
end;
```

```
begin   (* main body *)
  read_students;
  read_e1;
  read_e2;
  read_e3;
  textbackground(blue);
  repeat
    login(stud_index);
    if stud_index <> 0 then
      main_menu(stud_index)
  until false;
  readln
end.
```

Appendix 2 Working Schedule

Date	Task
March	Choice of Topic
April	Background research + Define the objectives
June	Design of Solution
Dec	Design + Implementation
Dec	Testing & Evaluation
Jan	Conclusion & Discussion + Final Report