Hong Kong Diploma of Secondary Education

Examination 2016

Information and Communication Technology

(Coursework)

Option D: Software Development

Title: Educational Software Program

Contents

Chapter 1 – Introduction

1.1 Background

A secondary school would like to design an educational software to assist students' learning. I am the IT project manager responsible for the project. I am going to provide solutions for the school.

There is no doubt that Information Technology can help student to learn better, which nowadays many students have a computer at home and use it for educational or entertainment.

With advancement of Information Technology, works can be done not only on paper, but also in computer. Complicated works such as calculating a massive equation can be done with the assist of computer. In this way, students can learn better and faster, which they can learn a lot of information in a small amount of time. Also, learning new knowledge through computer could be more fun if there are interaction between students and the program itself. Nowadays, teenagers love to play video games on computer, they playing it not only because of fun, but also because it can help them reduce stress from studying. Studying in Hong Kong can be extremely hard because there are a keen competition between students. So, by designing an interesting educational software for students, there is no doubt that it will help students learn better.

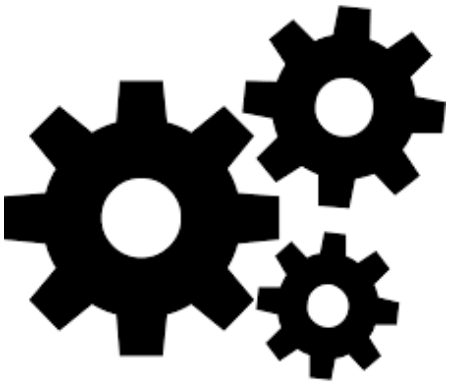
Chapter 2 - Design of Solution

2.1 Brief Description

In this project, I am going to develop a program to assist students' learning. In this program, it will be mainly focus on education of ICT, but modification can be done if they want to use it as teaching other subjects, such as English or Mathematics.

The educational software will be named as "eLearner". It will have the following functions:

i. Personal account for each students;
ii. Login function;
iii. Students can learn lessons about ICT;
iv. Students can do quizzes about ICT;
v. Students can play games related to ICT;



In this Chapter, I will design the program based on the functions I proposed in the above. I will design:

i. The overall structure of the program by refining the problem
ii. The formats of the data files for storing the user name and the password for each students
iii. The formats of the data files for storing lessons, both quiz' and game's questions and answer keys

I will assume that:

i. Students who are using the software should be studying in F.4 to F.6 and chosen ICT as their elective.

ii. There are total of 100 students who choose ICT as their elective.

iii. All the data files will store in D:\

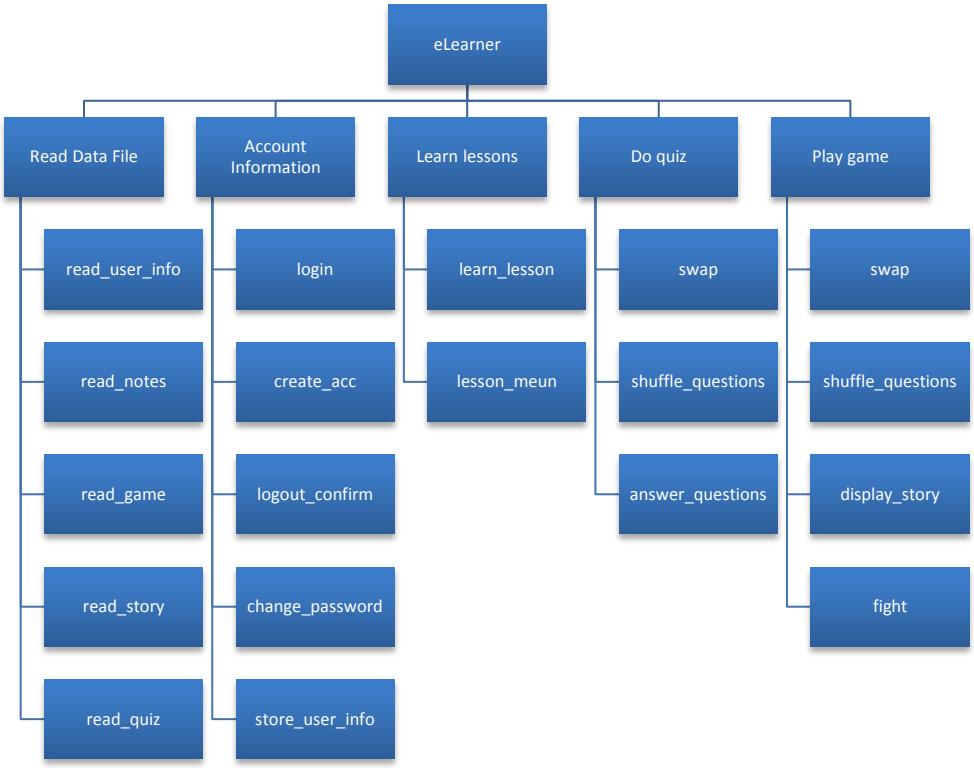
iv. There are 3 chapters in ICT curriculum

v. All questions in quiz are MC type

vi. All the MC questions contain 4 choices: A, B, C and D

2.2 Refinement of Problem

The problem above can be broken down into the following sub-problems.



For “Read data file”, as all the data file will be stored in D:\, the program need to read the information inside in order to process other modules. The program will read file that store user information, notes, quiz’s questions and answers story and game’s questions and answers.

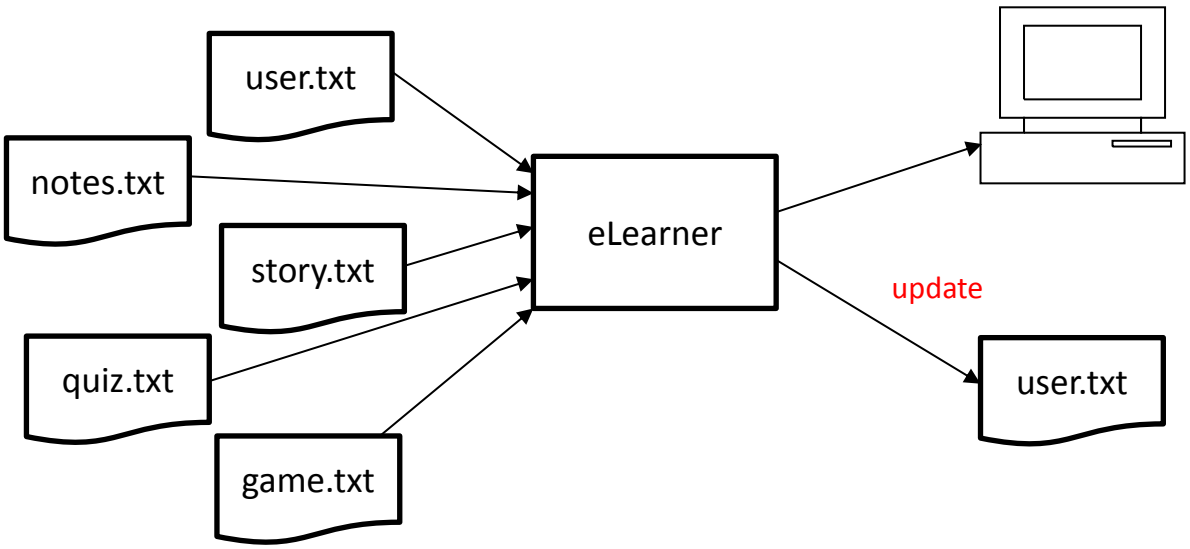
For “Account function”, this function let students to login their own account. Other than that, students can also create their own account if they don’t own one. While students are successfully login to their own account, they can even change their password if they are unsatisfied with the password. After that, the information will be stored.

For “Learn lessons”, this function can let students learn more knowledge though viewing the notes from the notes file. Students can choose from which chapter they would like to learn.

For “Do quiz”, this function provide a platform for students to challenge themselves. It will display MC type questions and choices. After students answered, it will indicate whether the student get it right or not. If the student answered the wrong answer, it will display the correct answer. At the end, it will also calculate the result after students finish the quiz.

For “Play game”, this function provide an entertainment way for students to challenge themselves. It is a RPG game where you need to restore peace to the village by killing demons. It will display short questions and require students to answer. If students answer a lot of questions wrongly, the game will over.

Data Flow:



2.3 Input Data File Formats

1. File storing user information:
   - One single file only
   - File name: user.txt
   - File type: plain text file
   - Data stored:

     The data file stores the record of each students, which includes the following information:

     - Students ID for each students (4 characters)
     - Password (4 characters)
     - Students' name (At most 25 characters)
   - File structure:

     Each line of the data file stores the record of one participant with the following format:

Students ID (4 characters)	Password (4 characters)	Students' name (At most 25 characters)
0001	1234	Au Sham Ki, Amy

- Sample File:

  user.txt

```
00011234Au Sham Ki, Amy
00021234Au Yue, Joanne
00031234Chan Kai Bong
00041234Chan Man Cheun
00051234Chan Mei Ling
00061234Chan Shui Wah, Shirley
00071234Chan Tai Man
00081234Chan Wai Yee, Wendy
00091234Chan Yick Yee, Eliza
00101234Chau Tung, Donnie
00111234Cheung Chi Chung
00121234Chan Tai Man
00131234Chu Kai Ki
00141234Chui Tse Lung
00151234Chung Kei Man, Mandy
00161234Fung Siu Wan, Melanie
00171234Ho Kam Shui, Tom
00181234Ho Li, Lily
00191234Hui Chung Hong
00201234Hung Yen Yen
00211234Kam Yung Kai
00221234Kwan Wan Cheong
00231234Kwan Yuen Ching, Cindy
00241234Kwok Foo Ho
00251234Lam Tik Man
00261234Lam Ting Cheong
00271234Lau Yiu Wing
00281234Lau Yum Meow
00291234Tang Tsz Kit
00301234Li Man Hon
00311234Tse Wai Tak
```

2. File storing lessons:
   - One single file only
   - File name: notes.txt
   - File type: plain text file
   - Data stored:

     The data file store each chapter's title and content.
   - File structure:

     First line represent which chapter

     Second line represent the title of the chapter

Third line represent the content of the chapter

- Sample file:

  notes.txt

```
Chapter 1
Basic Concepts of Input-Process-Output Cycle
A computer is a very powerful electronic machine. A computer can store a huge amount of data, perform complica
Chapter 2
Use of Stored Programs
A program is a sequence of commands. It should be provided to the computer in order to instruct the computer t
Chapter 3
Difference between Data and Information
Data is a collection of raw facts that are not organized and has no meaning on their own, while information ar
```

3. File storing quiz's information

   - One single file only
   - File name: quiz.txt
   - File type: plain text file
   - Data stored:

     The data store the quiz's questions and answer key

   - File structure:

     First line is the question

     Second to fifth line is the choices A-D for the question

     Sixth line is the answer for that question

   - Sample file:

```
Which of the following statements about computers and stored programs is correct?
A. Personal computers can complete a specific task themselves without stored programs or human instructions.
B. Stored programs can only be found in computers.
C. Computers can perform tasks described by stored programs.
D. All stored programs in a computer will be lost after the computer is shut down.
C
Which of the following statements is correct?
A. Data is a collection of facts in numerical form.
B. Information is meaningful to at least a specific group of people.
C. Computers can reject all incorrect data.
D. Data is used by computer operators while information is used by decision makers.
B
Which of the following is not a character coding system?
A. ASCII
B. Unicode
C. GB code
D. Bar code
D
Which of the following is not the major component in an information system?
A. Purposes
B. Technology
C. Personnel
D. Capital
D
Which of the following is not a feature of the Information Age?
A. There is an explosion of information.
B. There will be an increasing number of knowledge-based societies.
C. IT skills become much more important for people to get employed.
D. IT skills become much more important than language skills.
D
Which of the following is not a process in a washing machine?
```

4. File storing the story
    - One single file only
    - File name: story.txt
    - File type: plain text file
    - Data stored:

      Story for the RPG game
    - File structure:

      Each line represent each line of the story
    - Sample file:

      story.txt

```
Once upon a time,
There is a village where it is very peaceful.
Villagers are very happy and love this village where they are now living in.
But, one day,
A villager discovers that under the village, there is a temple.
Inside the temple, there is a shiny button.
Out of curiosity, the villager press the button.
Sadly, the villager don't know the button is to open the demon portal.
Now, the portal summon a lot of demon and destroy the village.
The King want you to kill the demon, destroy the portal and restore peace to the village.
So, the adventure begins.
```

5. File storing the game information
    - One single file only
    - File name: game.txt
    - File type: plain text file
    - Data stored:

      The game's questions and answer keys
    - File structure:

      First line is the question

      Second line is the answer to the question
    - Sample file

      game.txt

```
CPU
CENTRAL PROCESSING UNIT
ALU
ARITHMETIC AND LOGIC UNIT
RAM
RANDOM ACCESS MEMORY
ROM
READ ONLY MEMORY
USB
UNIVERSAL SERIAL BUS
VDU
VISUAL DISPLAY UNIT
ICT
INFORMATION AND COMMUNICATIONS TECHNOLOGY
PC
PERSONAL COMPUTER
HTML
HYPER TEXT MARKUP LANGUAGE
SQL
STRUCTURED QUERY LANGUAGE
ASCII
AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE
JS
JAVA SCRIPT
BMP
BITMAP
MP3
MPEG AUDIO LAYER 3
MP4
MPEG LAYER 4
ZIP
```

Chapter 3 – Implementation

3.1 Brief Description

- In this Chapter, I will discuss the implementation of the eLearner program.
- I will:
    1. Determine the data structures that will be used in the program.
    2. Describe the functions that will be performed by each procedure in the program.
    3. Explain the main algorithms used in the program
    4. Display some of the program codes
    5. Display the user interface

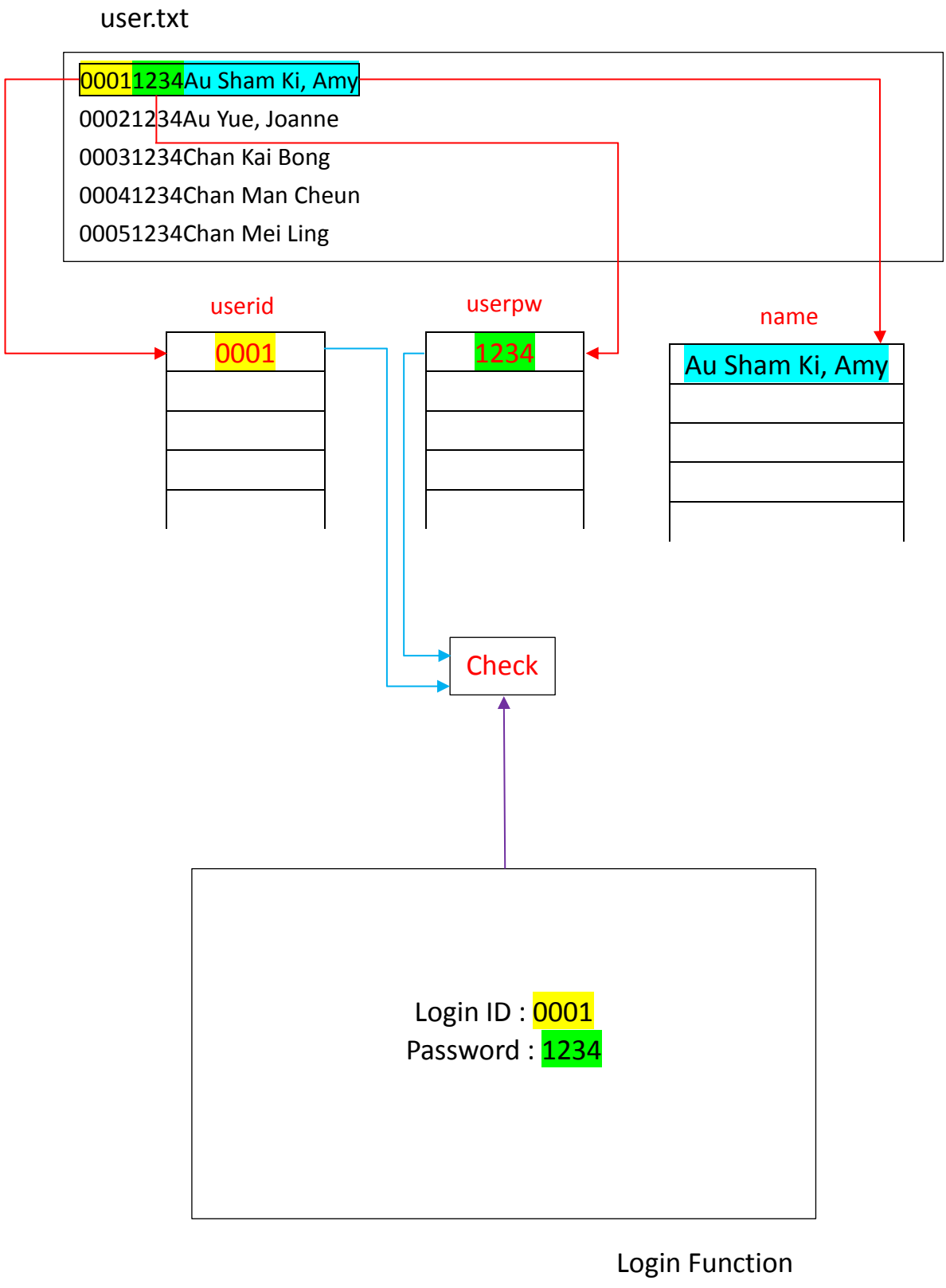
3.2 Data Structures

The following parallel arrays will be used to store the student ID, password and name of each students:

```
userid : array[1..max_user] of string[4];
userpw : array[1..max_user] of string[4];
name : array[1..max_user] of string[25];
```

where `max_user` is a constant defined at the beginning of the program. For example, `userid[1]`, `userpw[1]` and `name[1]` will store the corresponding data of the first students.

These array will be useful for providing an individual account for each students. For better illustration, following diagram in the next page will explain the function of this array.



Login Function

The following array will be used to store chapter title and content for each chapter:

```
chapter, title, content : array[1..total_chapter] of ansistring;
```

where `total_chapter` is a constant defined at the beginning of the program.
For example, `chapter[1] , title[1]` and `content[1]` will store the corresponding information of the first chapter. These array will be used for display lessons for students' study use.

The following array will be used to store quiz's question, choice and answer for each question in procedure `do_quiz`:

```
question, chA, chB, chC, chD, ans : array[1..50] of string;
```

For example, `question[1] , chA[1] , chB[1] , chC[1] , chD[1]` and `ans[1]` will store the corresponding information of the first chapter. The `ans[1]` array will store the answer of the corresponding question and it will be check with what the student have inputted to see if the student answer the question correctly or not, while the other arrays will store the question and choice of the corresponding question for display purpose.

The following array will be used to store game's story, question and answer for each question:

```
question, ans, story: array[1..50] of string;
```

For example, `question[1] , ans[1]` and `story[1]` will store the corresponding information of the first question and the story. The `story[1]` array will store a story line, which make the game more interesting by doing so. Other arrays will store the question and the answer, which will be check with students' input to decide whether the character student controlling attack or the monster attack. This make the game more challenging because students' character could die if students get too many question wrong.

3.3 Procedures in the Program

According to the sub-problems in Section 2.2 of Chapter 2, the following procedures will be constructed in the program:

1. Read Data File

    1.1. read_user_info

    Repeat to read the “user.txt” file until the end of the file and apply each column to arrays `userid[i]`, `userpw[i]` and `name[i]`.

    Number of times repeated equal to the number of users.

    1.2. read_notes

    Repeat to read the “notes.txt” file until the end of the file and apply each three column to arrays `chapter[i]`, `title[i]` and `content[i]` respectively.

    1.3. read_game

    Repeat to read the “game.txt” file until the end of the file and apply each two column to arrays `questions[i]` and `ans[i]` respectively.

    Number of times repeated equal to the total number of questions the game has.

    1.4. read_story

    Repeat to read the “story.txt” file until the end of the file and apply each column to array `story[i]`.

    Number of times repeated equal to the total number of lines the story has.

1.5. read_quiz

Repeat to read the “quiz.txt” file until the end of the file and apply each two column to arrays `questions[i]` and `ans[i]` respectively.

Number of times repeated equal to the total number of questions the quiz has.

2. Account information

2.1. login

To check whether the user input the correct UserID and password respectively. When the user input either one of them incorrectly, error message will show up to tell the user.

2.2. create_acc

Let user create an account if they don’t have one. It will require the user to input name, their own desired UserID and password. Also, it will check whether the name is more than 25 characters, the UserID and password are more than 4 characters and the UserID is same as any of the existence account. Error message will show up if the user do something wrong.

2.3. logout_confirm

Let the user to confirm whether they really want to logout or not. When the user logout, the “Thanks for using eLearner” message will show up and close the application afterward.

2.4. change_password

It will allow the user to change their password. It will first check the identity of the user by asking the user to input the old password again. And it will allow the user to change password

by asking the user to input the password twice to prevent mistype.

2.5. store_user_info

After a user register a new account or someone changed their password, store_user_info will be used to update the “user.txt” file.

3. Learn lessons

The procedure “lesson_meun” display a user interface to let user to choose which lesson they would like to learn. The procedure “learn_lesson” display lessons from the “notes.txt” file.

4. Do quiz

4.1. swap

Swap the questions in the “shuffle_questions” procedure.

4.2. shuffle_questions

Shuffle the questions in “quiz.txt” randomly.

4.3. answer_questions

Check whether the user get the correct answer for the quiz and calculate the result based on the user performance.

5. Play game

5.1. swap

Swap the questions in the “shuffle_questions” procedure.

5.2. shuffle_questions

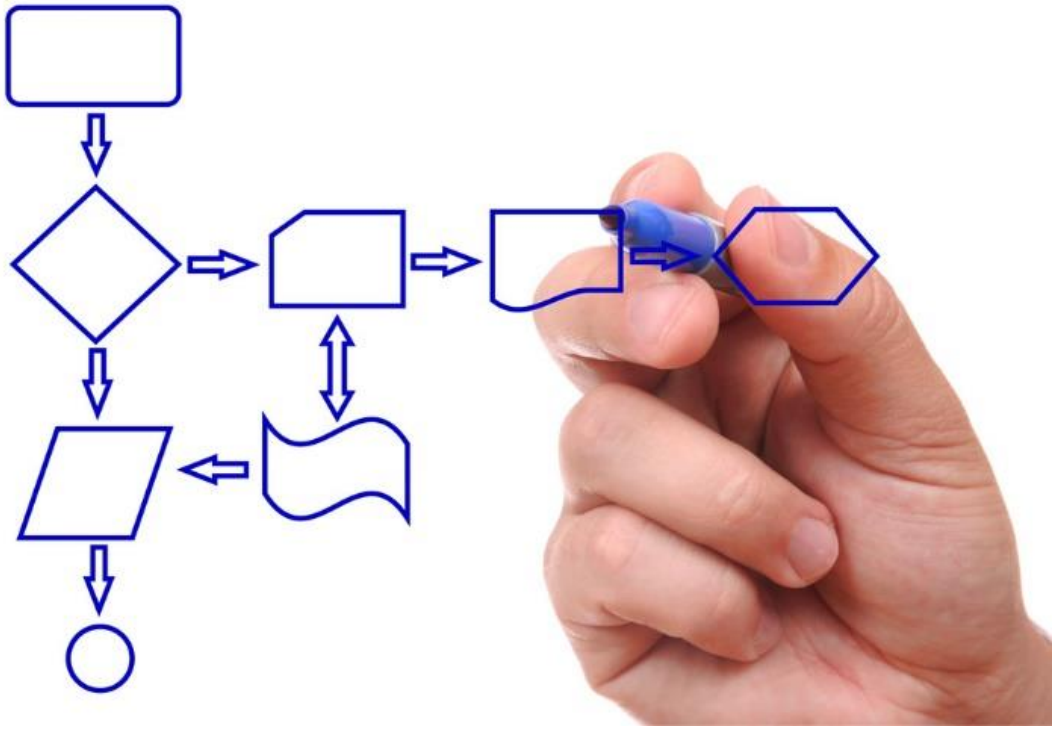
Shuffle the questions in “game.txt” randomly.

5.3. Display_story

Display story from the “story.txt” file.

5.4. fight

Check whether the user get the correct answer for the game and calculate the result based on the user performance, and display the game status afterward.

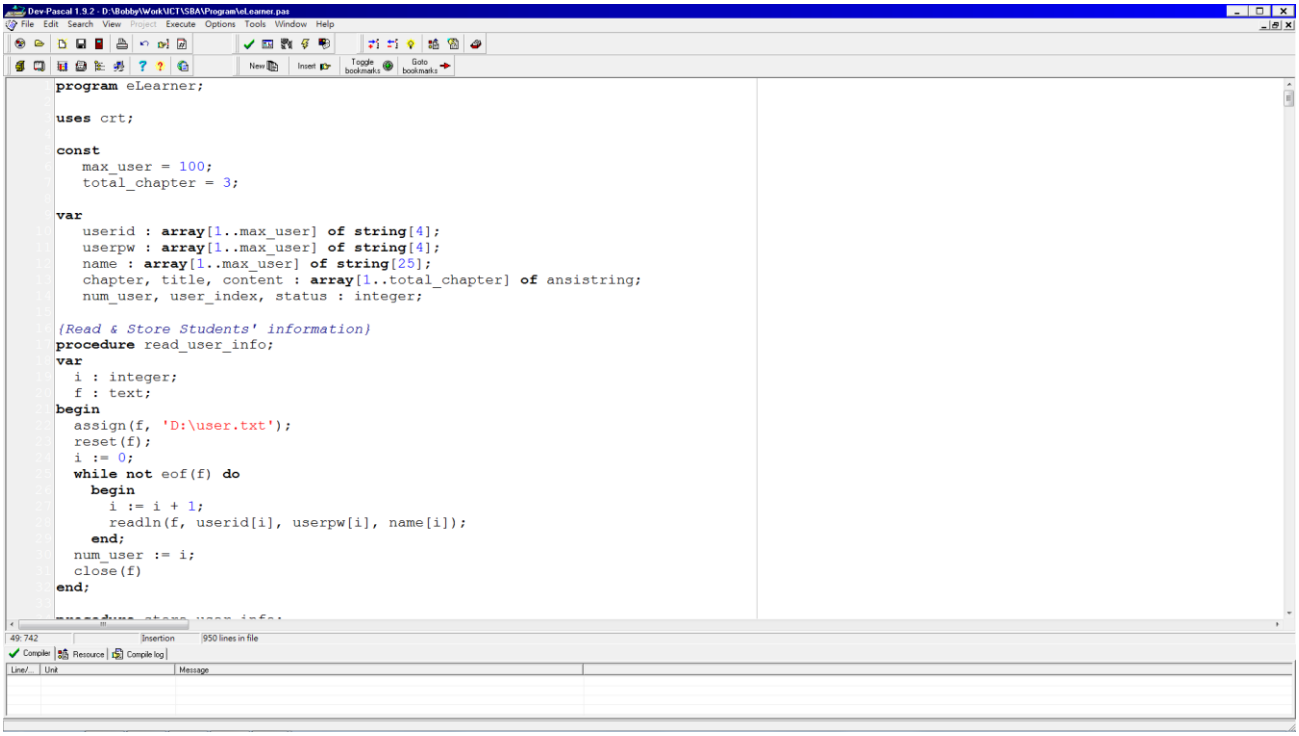


3.4 Program coding

I use Pascal as my programming language and use Dev-Pascal 1.9.2 as my tool for writing and compile the program.

I write the source program (eLeaner.pas) and then compile it into object program (eLearner.exe).

Refer to Appendix 1 for the complete program code.



```
program eLearner;

uses crt;

const
   max_user = 100;
   total_chapter = 3;

var
   userid : array[1..max_user] of string[4];
   userpw : array[1..max_user] of string[4];
   name : array[1..max_user] of string[25];
   chapter, title, content : array[1..total_chapter] of ansistring;
   num_user, user_index, status : integer;

{Read & Store Students' information}
procedure read_user_info;
var
  i : integer;
  f : text;
begin
  assign(f, 'D:\user.txt');
  reset(f);
  i := 0;
  while not eof(f) do
    begin
      i := i + 1;
      readln(f, userid[i], userpw[i], name[i]);
    end;
  num_user := i;
  close(f)
end;
```

↑ The tool I use for writing and compile the program.

3.5 Program Execution

1. Program file: eLearer.exe
2. Data file to be prepare for input:
   - ✓ The text file storing user information: user.txt
   - ✓ The text file storing lessons: notes.txt
   - ✓ The text file storing story: story.txt
   - ✓ The text file storing game questions: game.txt
   - ✓ The text file storing quiz questions: quiz.txt
   - ✓ All the program files and data files should be put into D:\
   - ✓ The files are prepared by Notepad++ for program testing. The data files can be edited with your own desired.

```
00011234Au Sham Ki, Amy
00021234Au Yue, Joanne
00031234Chan Kai Bong
00041234Chan Man Cheun
00051234Chan Mei Ling
00061234Chan Shui Wah, Shirley
00071234Chan Tai Man
00081234Chan Wai Yee, Wendy
00091234Chan Yick Yee, Eliza
00101234Chau Tung, Donnie
00111234Cheung Chi Chung
00121234Chan Tai Man
00131234Chu Kai Ki
00141234Chui Tse Lung
00151234Chung Kei Man, Mandy
00161234Fung Siu Wan, Melanie
00171234Ho Kam Shui, Tom
00181234Ho Li, Lily
00191234Hui Chung Hong
00201234Hung Yen Yen
00211234Kam Yung Kai
00221234Kwan Wan Cheong
00231234Kwan Yuen Ching, Cindy
00241234Kwok Foo Ho
00251234Lam Tik Man
00261234Lam Ting Cheong
00271234Lau Yiu Wing
00281234Lau Yum Meow
00291234Tang Tsz Kit
00301234Li Man Hon
00311234Tse Wai Tak
```

user.txt

```
Chapter 1
Basic Concepts of Input-Process-Output Cycle
A computer is a very powerful electronic machine. A computer can store a huge amount of data, perform complica
Chapter 2
Use of Stored Programs
A program is a sequence of commands. It should be provided to the computer in order to instruct the computer t
Chapter 3
Difference between Data and Information
Data is a collection of raw facts that are not organized and has no meaning on their own, while information ar
```

notes.txt

```
Once upon a time,
There is a village where it is very peaceful.
Villagers are very happy and love this village where they are now living in.
But, one day,
A villager discovers that under the village, there is a temple.
Inside the temple, there is a shiny button.
Out of curiosity, the villager press the button.
Sadly, the villager don't know the button is to open the demon portal.
Now, the portal summon a lot of demon and destroy the village.
The King want you to kill the demon, destroy the portal and restore peace to the village.
So, the adventure begins.
```

story.txt

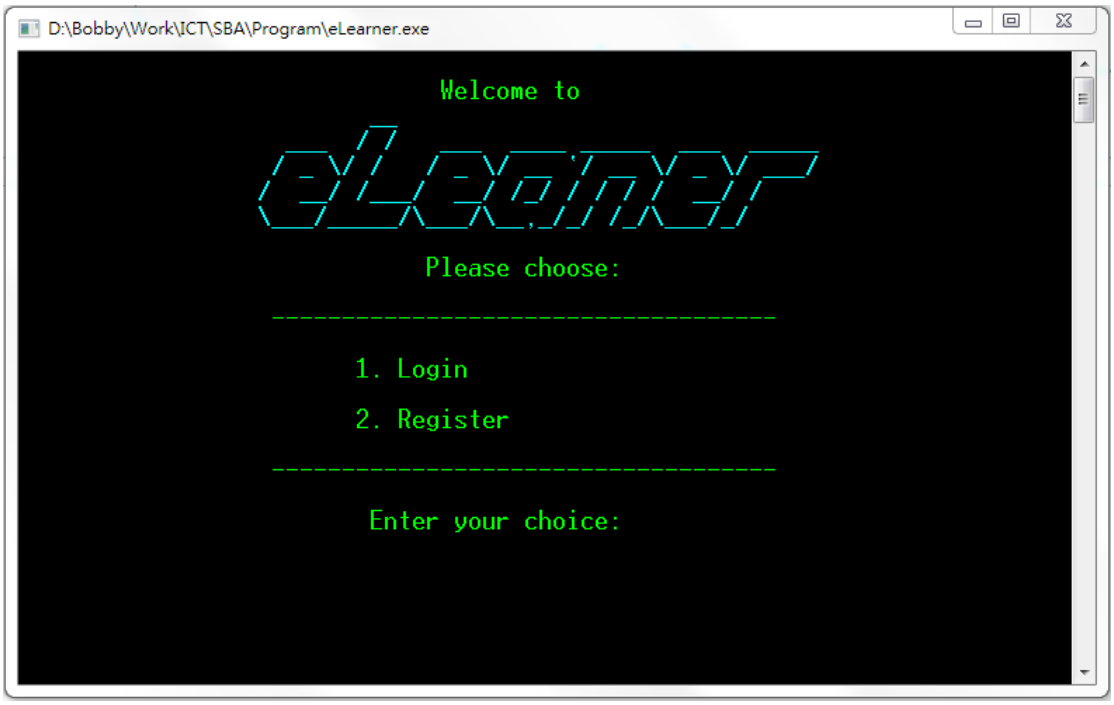
```
CPU
CENTRAL PROCESSING UNIT
ALU
ARITHMETIC AND LOGIC UNIT
RAM
RANDOM ACCESS MEMORY
ROM
READ ONLY MEMORY
USB
UNIVERSAL SERIAL BUS
VDU
VISUAL DISPLAY UNIT
ICT
INFORMATION AND COMMUNICATIONS TECHNOLOGY
PC
PERSONAL COMPUTER
HTML
HYPER TEXT MARKUP LANGUAGE
SQL
STRUCTURED QUERY LANGUAGE
ASCII
AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE
JS
JAVA SCRIPT
BMP
BITMAP
MP3
MPEG AUDIO LAYER 3
MP4
MPEG LAYER 4
ZIP
```

game.txt

```
Which of the following statements about computers and stored programs is correct?
A. Personal computers can complete a specific task themselves without stored programs or human instructions.
B. Stored programs can only be found in computers.
C. Computers can perform tasks described by stored programs.
D. All stored programs in a computer will be lost after the computer is shut down.
C
Which of the following statements is correct?
A. Data is a collection of facts in numerical form.
B. Information is meaningful to at least a specific group of people.
C. Computers can reject all incorrect data.
D. Data is used by computer operators while information is used by decision makers.
B
Which of the following is not a character coding system?
A. ASCII
B. Unicode
C. GB code
D. Bar code
D
Which of the following is not the major component in an information system?
A. Purposes
B. Technology
C. Personnel
D. Capital
D
Which of the following is not a feature of the Information Age?
A. There is an explosion of information.
B. There will be an increasing number of knowledge-based societies.
C. IT skills become much more important for people to get employed.
D. IT skills become much more important than language skills.
D
Which of the following is not a process in a washing machine?
```

quiz.txt

3. User interface of the program (The input screen)



The program will provide instruction for what to input.

Chapter 4 Testing & Evaluation

4.1. Brief Description

- To find out the bugs (logical and run-time errors) in the program,
- To check whether the program can achieve its purposes,
- To see whether the program is user-friendly or not,
- To debug and improve the program based on the testing and evaluation results

4.2. Testing and Evaluation Plan

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

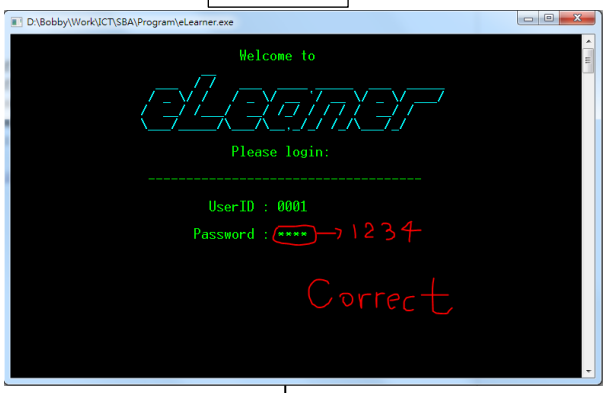
- Internal testing and evaluation / Tested and evaluated by myself:
  - The program will be tested intensively by myself – the programmer
  - I will prepare different test cases to test the program thoroughly
  - The test cases include some correct input data (from files & keyboard) with known results for checking the correctness of the program, some incorrect input data to see whether the program can handle invalid input reasonably, etc.
  - I will also evaluate the programs according to its user-friendless, performance, flexibility for future development, reusability of program codes, etc.

4.3. Internal Testing

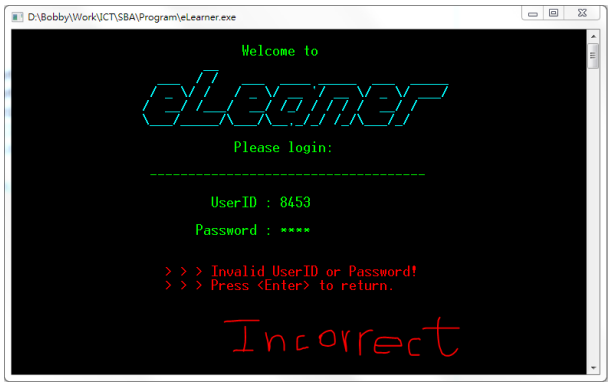
Test case 1

Purpose:	To check whether the login system working or not by using correct and incorrect data
Input:	Correct UserID and password Incorrect UserID and password Incorrect UserID and correct password Correct UserID and incorrect password
Expected Output:	Expect using the correct UserID and password to login successfully, other set of data will result in an error message show up
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	None

Correct



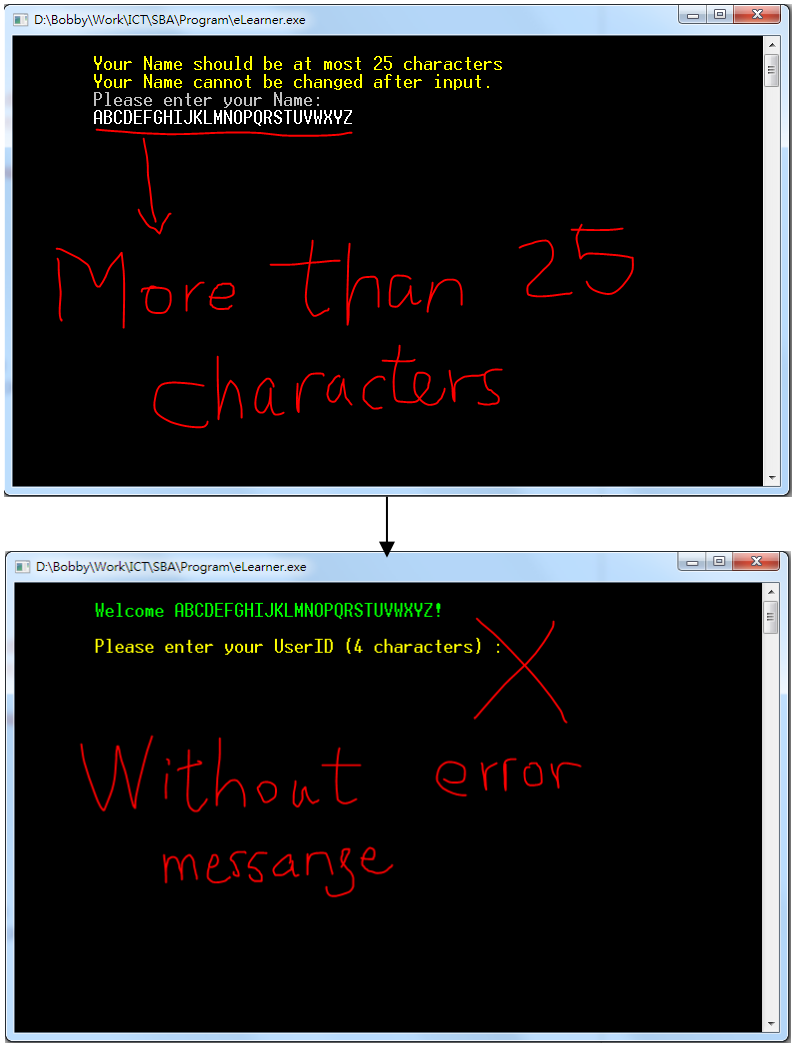
Incorrect



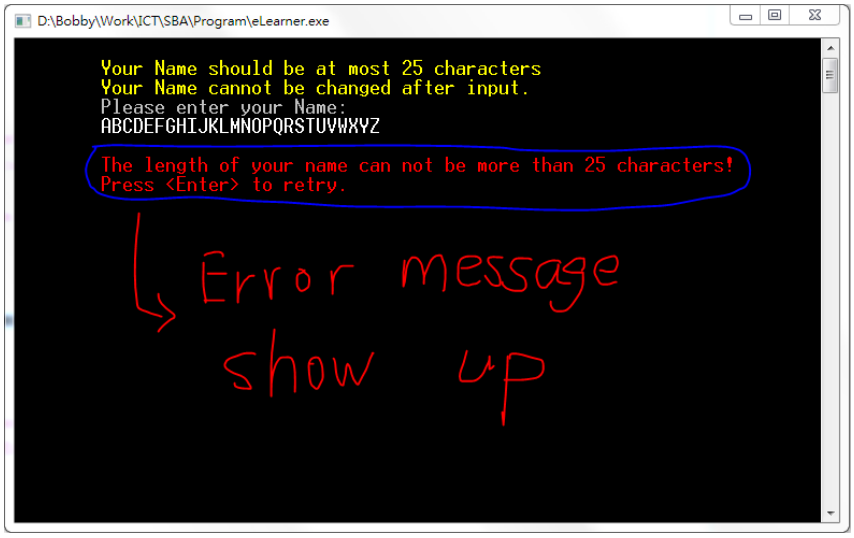
Test case 2

Purpose:	To check whether the register work or not by using correct and incorrect data.
Input:	Normal data with name, UserID and password Wrong data excess the number of characters it allowed
Expected Output:	By using normal data, a new account will be created By using wrong data, error message will show up
Actual Output:	Actual results are the same as the expected results, expect the wrong name data.
Test Result:	The program did not response as expected
Follow-up Action:	Modify the program so that it will check whether the name is excessing 25 characters or not. Repeat-loop is used.

Origin response of program:



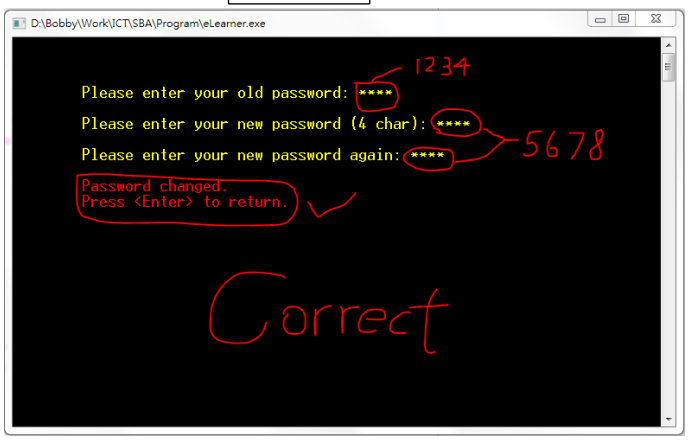
Response of program after modification:



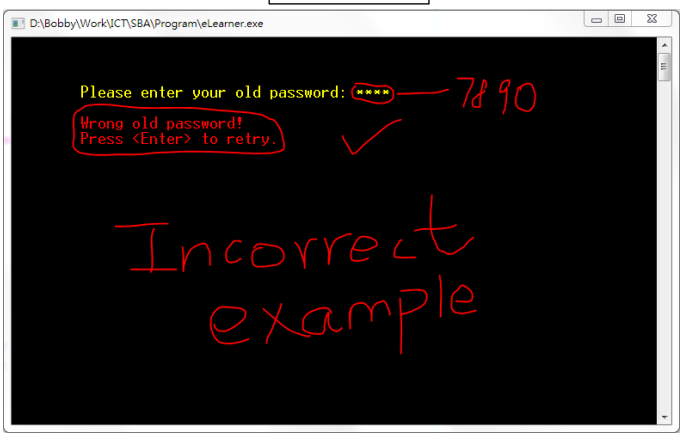
Test case 3

Purpose:	To check whether the changing password system work or not by using correct and incorrect data
Input:	Correct data with old password and new password Incorrect data with wrong old password and different new password
Expected Output:	By using correct data, password will be changed By using incorrect data, error message show up
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / no bugs found
Follow-up Action:	None

Correct



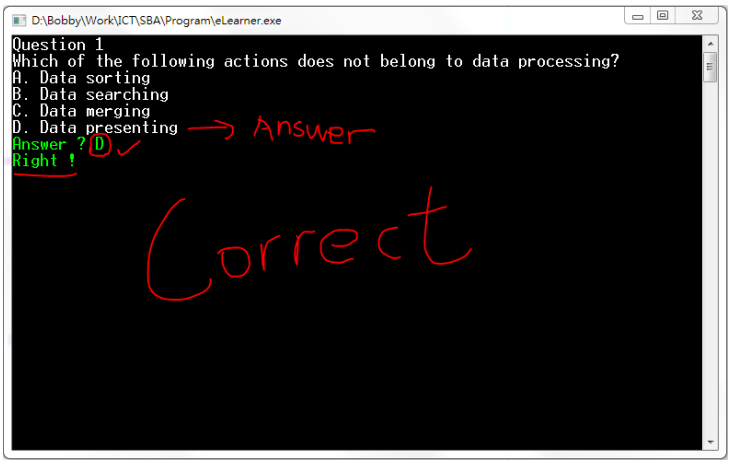
Incorrect



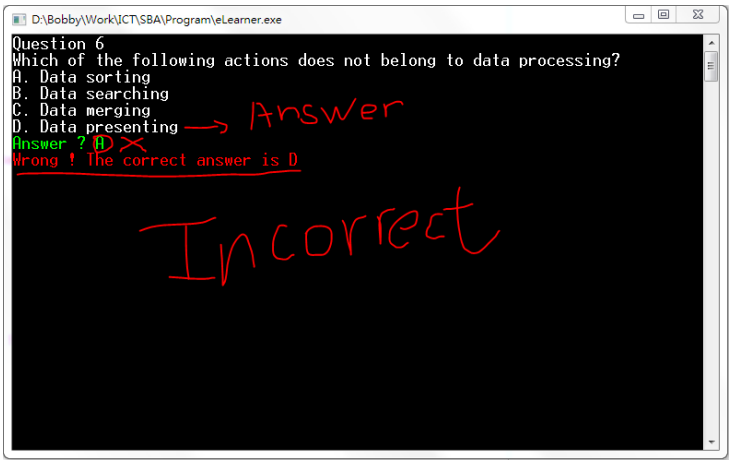
Test case 4

Purpose:	To check whether the quiz system work or not by answering correct, incorrect and abnormal answer
Input:	Correct answer Incorrect answer Abnormal answer
Expected Output:	By input correct answer, it will say that you are right. By input incorrect answer, it will say that you are wrong. By input abnormal answer, it will ask you to input a normal answer
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	None

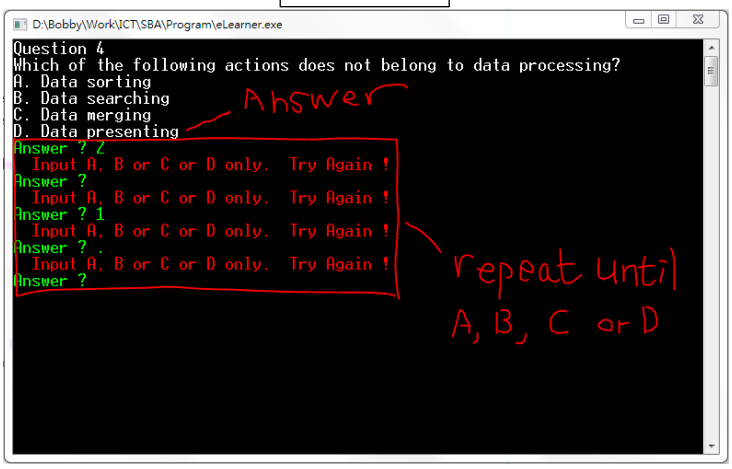
Correct



Incorrect



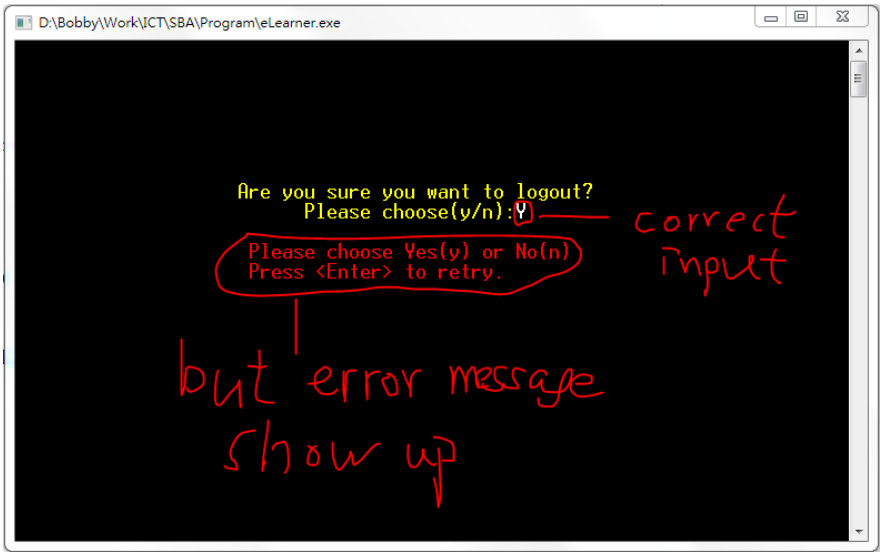
Abnormal



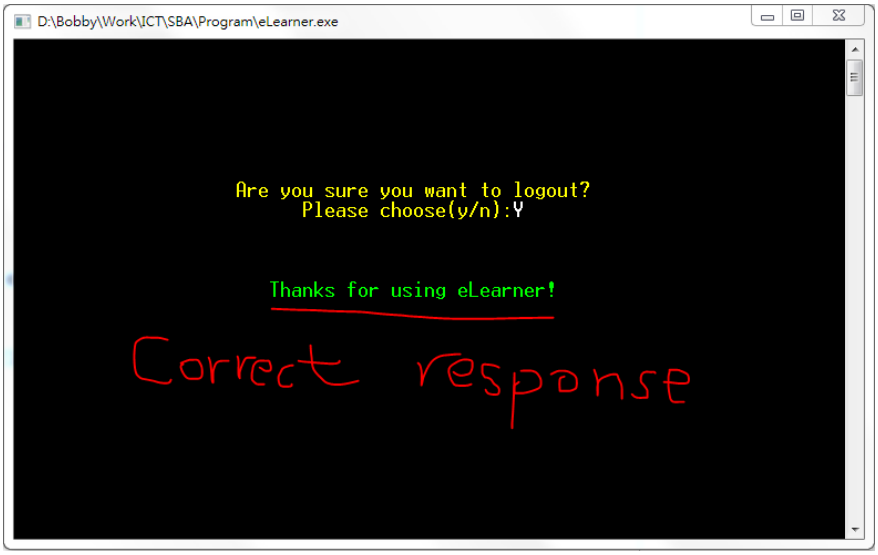
Test case 5

Purpose:	To check whether the logout system work or not by using correct and incorrect data
Input:	Correct data including uppercase letter and lower case letter Incorrect data that is not “y” or ”n”
Expected Output:	User will be logout successfully when using correct data Error message will show up when using incorrect data
Actual Output:	Actual results are the same as the expected results, expect error message will show up when using the uppercase letter correct data
Test Result:	The program did not response as expected
Follow-up Action:	Modify the program so that it also accept upper case letter

Origin response of program:



Response of program after modification:



4.4. Self-Evaluation

The program provide a platform for students have way to learn new things or challenge themselves with quiz or games. But sadly the interface may not be appearing to all the students as it is lack of color choice in Pascal and hard to present graphic with Pascal. Also we don't have a way to ensure that every students who use this educational system learn the same amount of stuff or did they really learn it or not. So, I think it is the good features and shortcomings of my own program.

Chapter 5 Conclusion & Discussion

1. Pros and cons of my Program

I have spent lots of time in doing this educational system. From planning, making and testing, there are lots of ideas put in it and I tried to make it perfect. Although it is corrected lots of times, still there are something not really convenient for students, which I cannot correct. The following will be showing my education system's pros and cons.

The sample of my own educational system focus on educating students who choose ICT as their elective subjects. So, ICT related knowledge is needed in order to finish the programs. Although the sample is focus on ICT itself, the program can change to other subjects to focus on by just editing the data file. It is very easy to edit because most of the resources are in the data file. After editing the data file, the program will automatically change the focusing subject. So, teachers can easily reuse the program coding by just editing the data file itself, not by editing the source code, as most of the teachers may not have any idea about how to edit the source code using Pascal programming language. Other than that, the software include a place for students to learn lessons, do quiz and play game. Both learn lessons and do quiz can improve students' knowledge towards the specific subject, where play game can also improve students' knowledge, it can also develop different skill from playing game, for example, eye-hand coordination and decision-making skill. I believe that students learn at least something

from this software.

On the other hand, the program can't ensure every students learning the same amount of knowledge. Some of the students may just play game and find the answer on the Internet as it is easy to find these answer when students Google it on the Internet. But that is something I can't improve as the technology nowadays is very advanced. Also, when someone edit the data file to include more questions or lessons and it became longer, the loading time will become quite slow for low-end computer user. I think students would not really mind whether it is fast or not, but for some of the students whose can't afford a computer with powerful CPU, their loading time may be longer. I am sincerely sorry about that.

I hope I can make it more perfect next time, in order to make my educational software smoother and suitable for all students in Hong Kong, or even the World.

2. Future Improvement

In future, I really hope I can add more function into this educational software. I would try to add a menu where you can choose different subjects, so that it will not be a mess when there are too many different version of "eLearner" of different subjects. If there were a chances, I would also try to use other different software to create an interesting game for students or even kid to play with, because nowadays students in Hong Kong have a lot of pressure from either school or parents. With an interesting game to relax themselves from the horrible amount for extracurricular activity, they can perform better when there are not that

many stress around them. In this way, students can learn more effectively and be able to perform really well in the exam.

3. Self-Reflection

In making this educational software, I have learnt more about ICT and software development. Most important is how ones software to be develop and test. My dream is to be a game developer and before making this software, I'm a little bit scared of my lack of experience to software development. But after making this software, it give me confident to face the challenges when being a game developer.

Also, I have learnt how to collect useful data and ways to get information. These are very useful for me. Furthermore, I learnt how to create a software report, which is also very useful for me because in my future career, I may need to create one for another software, or one for the company I am working in. But these are just dreams that I need to be achieve in the future. I feel very confident about it as by making this program, it give me experience to make a software.

Chapter 6 Reference and Acknowledgement

From Internet website

1. https://www.google.com.hk
2. http://patorjk.com/software/taag/#p=display&f=Graffiti&t=Type%20Something%20
3. http://www.chris.com/ascii/
4. http://www.retrojunkie.com/asciiart/
5. http://computer-programming-forum.com/29-pascal/7af4f3f05f738777.htm
6. http://www.irietools.com/iriepascal/progref298.html

From books

1. NSS Information and Communication Technology D1
2. NSS Information and Communication Technology D2
3. NSS Information and Communication Technology Compulsory 1
4. Some other report samples

Acknowledgement

1. Thanks Mr. Chu for his help

Appendices

Appendix 1 – Program Code (after Testing & Evaluation)

```
program eLearner;

uses crt;

const
   max_user = 100;
   total_chapter = 3;

var
   userid : array[1..max_user] of string[4];
   userpw : array[1..max_user] of string[4];
   name : array[1..max_user] of string[25];
   chapter, title, content : array[1..total_chapter] of ansistring;
   num_user, user_index, status : integer;

{Read & Store Students' information}
procedure read_user_info;
var
  i : integer;
  f : text;
begin
  assign(f, 'D:\user.txt');
  reset(f);
  i := 0;
  while not eof(f) do
    begin
      i := i + 1;
      readln(f, userid[i], userpw[i], name[i]);
    end;
  num_user := i;
  close(f)
end;
```

```
procedure store_user_info;
var
  i : integer;
  f : text;
begin
  assign(f, 'D:\user.txt');
  rewrite(f);
  for i := 1 to num_user do
    begin
      writeln(f, userid[i], userpw[i], name[i]);
    end;
  close(f)
end;

{Read notes file}
procedure read_notes;
var
  i : integer;
  f : text;
begin
  assign(f, 'D:\notes.txt');
  reset(f);
  i := 0;
  while not eof(f) do
    begin
      i := i + 1;
      readln(f, chapter[i]);
      readln(f, title[i]);
      readln(f, content[i]);
    end;
  close(f)
end;

{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;

{Play game}
procedure play_game;
var
  pointer, totalQ, totalS, hit, health, attack : integer;
  question, ans, story: array[1..50] of string;

  {Read game questions file}
  procedure read_game;
  var
    i : integer;
    f : text;
  begin
    assign(f, 'D:\game.txt');
    reset(f);
    i := 0;
```

```
  while not eof(f) do
    begin
      i := i + 1;
      readln(f, question[i]);
      readln(f, ans[i]);
    end;
  totalQ := i;
  close(f)
end;

{Read story file}
procedure read_story;
var
  i : integer;
  f : text;
begin
  assign(f, 'D:\story.txt');
  reset(f);
  i := 0;
  while not eof(f) do
    begin
      i := i + 1;
      readln(f, story[i]);
    end;
  totalS := i;
  close(f);
end;

procedure swap(var s1, s2 : string);
var
  tmp : string;
begin
  tmp := s1;
  s1 := s2;
  s2 := tmp
end;

{Shuffle Questions}
```

```
procedure shuffle_questions;
var
  i, k, p : integer;
begin
  randomize;
  for i := 1 to totalQ do
    begin
      k := random(totalQ)+1;
      p := random(totalQ)+1;
      swap(question[k],question[p]);
      swap(ans[k],ans[p])
    end;
end;

procedure display_story;
var
  i : integer;
begin
   clrscr;
   for i := 1 to totalS do
     begin
       textcolor(14);
       writeln(story[i]);
       readln;
     end;
end;

procedure screen_idle;
begin
   textcolor(15);
   writeln('      !      ');
   writeln('     .-.     ');
   writeln('   __|=|__    ');
   writeln('  (_/`-`\_)   ');
   writeln('  //\___/\\   ');
   writeln('  <>/   \<>   ');
   writeln('   \|_._|/    ');
   writeln('    <_I_>     ');
```

```
   writeln('     |||     ');
   writeln('    /_|_\    ');
   writeln('-----------------------------------------------');
end;

procedure screen_first;
begin
   textcolor(15);
   writeln('                                      ');
   writeln('      !                               ');
   writeln('     .-.                              ');
   writeln('   __|=|__             (_(            ');
   writeln('  (_/`-`\_)           (**)            ');
   writeln('  //\___/\\          _  "\ )>,_    .--> ');
   writeln('  <>/   \<>         _>--w/((_ >,_.@    ');
   writeln('   \|_._|/               ///          ');
   writeln('    <_I_>                "`"          ');
   writeln('     |||                               ');
   writeln('    /_|_\                             ');
   writeln('-----------------------------------------------');
end;

procedure screen_second;
begin
   textcolor(15);
   writeln('                             v      ');
   writeln('      !                (__)v | v    ');
   writeln('     .-.               /\/\\_|_/    ');
   writeln('   __|=|__             _\__/  |     ');
   writeln('  (_/`-`\_)           /  \/`\<`)    ');
   writeln('  //\___/\\           \ (  |\_/     ');
   writeln('  <>/   \<>          /)))-(  |     ');
   writeln('   \|_._|/           / /^ ^ \ |     ');
   writeln('    <_I_>           /  )^/\^( |     ');
   writeln('     |||           )_//`__>> |     ');
   writeln('    /_|_\             #   #`  |     ');
   writeln('-----------------------------------------------');
end;
```

```
procedure screen_final;
begin
   textcolor(15);
   Writeln('                                        ');
   writeln('                                        ');
   writeln('                                        ');
   writeln('                                _______     ');
   writeln('      !                        /    \\\\    ');
   writeln('     .-.                      /  /\  \\\\   ');
   writeln('   __|=|__                    /  ///\  \\\\  ');
   writeln('  (_/`-`\_)                  /  //// \__\\\\ ');
   writeln('  //\___/\\                   \  \\\\ /  //// ');
   writeln('  <>/   \<>                  \  \\\\  ////  ');
   writeln('   \|_._|/                    \  \\\\////   ');
   writeln('    <_I_>                      \  \\\///    ');
   writeln('     |||                        `""""""`    ');
   writeln('    /_|_\                                  ');
   writeln('-----------------------------------------------');
end;

procedure instructions;
begin
  clrscr;
  writeln;
  textcolor(13);
  writeln('  Instructions');
  writeln('  Please answer the following questions.');
  writeln('  Each time you answer correctly, you will attack.');
  writeln('  But if you answer it wrong, the monster will attack you.');
  writeln('  You have total of 4 heart');
  writeln('  When you have 0 heart, game over.');
  writeln('  Press <Enter> to start.');
  readln;
  writeln
end;

procedure fight;
```

```
var
  answer : string;
begin
  begin
    textcolor(7);
    writeln('Q', pointer);
    writeln('What is the "', question[pointer],'" full name?');
    textcolor(10);
     readln(answer);
    answer := upcase(answer);
    if answer = ans[pointer] then
       begin
        textcolor(14);
        writeln('  Right !');
        writeln('  You attacked the monster!');
        hit := hit - 1;
        writeln('  The monster have ', hit, ' heart left.');
        readln;
      end
    else
      begin
        textcolor(12);
        writeln('  Wrong ! The correct answer is ', ans[pointer]);
        writeln('  The monster attacked you for ', attack, ' health!');
        health := health - attack;
        writeln('  You have ', health,' heart left.');
        readln;
      end;
    writeln;
   end;
 end;

procedure idle;
begin
  clrscr;
  screen_idle;
  textcolor(14);
  writeln('  Press <Enter> to continue.');
```

```
    readln;
  end;

  procedure gameover;
  begin
    clrscr;
    textcolor(12);
    writeln('  You died!');
    writeln('  Game Over!');
    readln;
  end;

begin
  pointer := 0;
  health := 4;
  read_game;
  read_story;
  shuffle_questions;
  display_story;
  instructions;
  idle;
  hit := 1;
  attack := 1;
  textcolor(12);
  writeln('  You encounter Little Demon!');
  readln;
  repeat
    pointer := pointer + 1;
    clrscr;
    screen_first;
    fight;
  until (hit = 0) or (health = 0);
  if hit = 0 then
    begin
      idle;
      hit := 2;
      attack := 2;
      health := 4;
```

```
      textcolor(14);
      writeln('  Your are fully healed!');
      textcolor(12);
      writeln('  You encounter Devil Ambassador!');
      readln;
        repeat
          pointer := pointer + 1;
          clrscr;
          screen_second;
          fight;
        until (hit = 0) or (health = 0);
        if hit = 0 then
          begin
            idle;
            hit := 3;
            attack := 2;
            health := 4;
            textcolor(14);
            writeln('  Your are fully healed!');
            textcolor(12);
            writeln('  You encounter the Demon Portal!');
            readln;
              repeat
                pointer := pointer + 1;
                clrscr;
                screen_final;
                fight;
              until (hit = 0) or (health = 0);
              if hit = 0 then
                begin
                  idle;
                  textcolor(14);
                  writeln('  You successfully destroy the portal and restore
peace to the village!');
                  writeln('  Congratulation!');
                end
              else
                gameover;
```

```
          end
        else
          gameover;
    end
  else
    gameover;
  writeln;
  textcolor(10);
  write('Press <Enter> to return. ');
  status := 2;
  readln
end;

{Do quiz}
procedure do_quiz;
var
  totalQ, correct : integer;
  question, chA, chB, chC, chD, ans : array[1..50] of string;

  {Read quiz file}
  procedure read_quiz;
  var
    i : integer;
    f : text;
  begin
    assign(f, 'D:\quiz.txt');
    reset(f);
    i := 0;
    while not eof(f) do
      begin
        i := i + 1;
        readln(f, question[i]);
        readln(f, chA[i]);
        readln(f, chB[i]);
        readln(f, chC[i]);
        readln(f, chD[i]);
        readln(f, ans[i]);
      end;
```

```
  totalQ := i;
  close(f)
end;

procedure swap(var s1, s2 : string);
var
  tmp : string;
begin
  tmp := s1;
  s1 := s2;
  s2 := tmp
end;

{Shuffle Questions}
procedure shuffle_questions;
var
  i, k, p : integer;
begin
  randomize;
  for i := 1 to totalQ do
    begin
      k := random(totalQ)+1;
      p := random(totalQ)+1;
      swap(question[k],question[p]);
      swap(chA[k],chA[p]);
      swap(chB[k],chB[p]);
      swap(chC[k],chC[p]);
      swap(chD[k],chD[p]);
      swap(ans[k],ans[p]);
    end;
end;

procedure answer_questions;
var
  answer : char;
  i : integer;
begin
  clrscr;
```

```
textcolor(14);
writeln;
writeln('  Instructions:');
writeln('  There are ', totalQ, ' questions in this quiz.');
readln;
writeln;
correct := 0;
for i := 1 to totalQ do
  begin
    clrscr;
    textcolor(15);
    writeln('Question ', i);
    writeln(question[i]);
    writeln(chA[i]);
    writeln(chB[i]);
    writeln(chC[i]);
    writeln(chD[i]);
    repeat
      textcolor(10);
       write('Answer ? ');
       readln(answer);
      answer := upcase(answer);
      if not (answer in ['A', 'B', 'C', 'D']) then
        begin
          textcolor(12);
           writeln('  Input A, B or C or D only.  Try Again !')
        end;
    until answer in ['A', 'B', 'C', 'D'];
    if answer = ans[i] then
      begin
       textcolor(10);
        writeln('Right !');
       readln;
        correct := correct + 1
      end
    else
      begin
        textcolor(12);
```

```
          writeln('Wrong ! The correct answer is ', ans[i]);
          readln;
          writeln
        end;
     end;
    textcolor(14);
    writeln('  Your correct percentage is ', correct*100/totalQ:5:2,
'%');
    writeln('  You answer ', correct, ' question(s) correctly out of ',
totalQ, ' questions.');
    readln;
   end;

begin
  read_quiz;
  shuffle_questions;
  answer_questions;
  textcolor(12);
  writeln;
  write('  Press <Enter> to return. ');
  status := 2;
  readln
end;

{Display lesson}
procedure learn_lesson(var i : integer);
begin
  clrscr;
  textcolor(14);
  writeln(chapter[i]);
  readln;
  writeln(title[i]);
  readln;
  textcolor(15);
  writeln(content[i]);
  readln;
  textcolor(12);
  writeln;
```

```
  writeln;
  writeln('Lessons finished! Press <Enter> to return');
  status := 3;
  readln;
end;

{Lessons Menu}
procedure lesson_menu;
var choice : integer;
begin
  clrscr;
  writeln;
  textcolor(11);
  writeln('                        __                          ');
  writeln('                ___  / /   ___  ____ _____  ___  _____');
  writeln('               / _ \/ /   / _ \/ __ `/ __ \/ _ \/ ___/');
  writeln('              /  __/ /___/  __/ /_/ / / / /  __/ /    ');
  writeln('              \___/_____/\___/\__,_/_/ /_/\___/_/     ');
  writeln;
  textcolor(10);
  writeln('                        Choose lessons:');
  writeln;
  writeln('               ------------------------------------');
  writeln;
  writeln('                  1.', chapter[1]);
  writeln;
  writeln('                  2.', chapter[2]);
  writeln;
  writeln('                  3.', chapter[3]);
  writeln;
  writeln('                  4.Return to main menu');
  writeln;
  writeln('               ------------------------------------');
  writeln;
  write('                 Enter your choice: ');
  textcolor(14);
  readln(choice);
  writeln;
```

```
  if choice <> 4 then
    learn_lesson(choice)
  else
    status := 2;
end;

{Changing Students' Password}
procedure change_password(user_index : integer);
var
  oldpass, newpass1, newpass2 : string;
  pwchanged : boolean;
begin
  pwchanged := false;
  repeat
    clrscr;
    writeln;
    writeln;
    writeln;
    textcolor(14);
    write('       Please enter your old password: ');
    oldpass := GetPword;
    if oldpass <> userpw[user_index] then
      begin
        textcolor(12);
        writeln;
        writeln('       Wrong old password!');
        write('       Press <Enter> to retry. ');
        readln
      end
    else
      begin
        textcolor(14);
        writeln;
        write('       Please enter your new password (4 char): ');
        newpass1 := GetPword;
        if length(newpass1) <> 4 then
          begin
            textcolor(12);
```

```
          writeln;
          writeln('       The length of password must be 4!');
          write('       Press <Enter> to retry. ');
          readln
        end
      else
        begin
          textcolor(14);
          writeln;
          write('       Please enter your new password again: ');
          newpass2 := GetPword;
          if newpass1 <> newpass2 then
            begin
              textcolor(12);
              writeln;
              writeln('       The new password does not match!');
              write('       Press <Enter> to retry. ');
              readln
            end
          else
            begin
              userpw[user_index] := newpass1;
              store_user_info;
              pwchanged := true;
              status := 2;
              textcolor(12);
              writeln;
              writeln('       Password changed.');
              write('       Press <Enter> to return. ');
              readln
            end
        end
    end
  until pwchanged
end;

procedure logout_confirm;
var
```

```
  choice : char;
  temp : integer;
  OK : boolean;
begin
  repeat
    clrscr;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    textcolor(14);
    writeln('                 Are you sure you want to logout?');
    write('                     Please choose(y/n):');
    textcolor(15);
    readln(choice);
    temp := ord(choice);
    if (temp = 110) or (temp = 121) or (temp = 78) or (temp = 89) then
      begin
        OK := true;
        if (temp = 89) or (temp = 121) then
          status := 0
        else
          status := 2
      end
    else
      begin
        writeln;
        textcolor(12);
        writeln('                  Please choose Yes(y) or No(n)');
        write('                 Press <Enter> to retry. ');
        readln
      end
  until OK;
end;
```

```
procedure main_menu(user_index : integer);
var
  choice : integer;
begin
  clrscr;
  writeln;
  textcolor(11);
  writeln('                          __                              ');
  writeln('                 ___  / /   ___  ____ _____  ___  _____');
  writeln('                / _ \/ /   / _ \ __ `/ __ \/ _ \ ___/');
  writeln('               /  __/ /___/  __/ /_/ / / / /  __/ /    ');
  writeln('               \___/_____/\___/\__,_/_/ /_/\___/_/     ');
  writeln;
  textcolor(10);
  writeln('                              User Menu');
  writeln;
  writeln('                ------------------------------------');
  writeln;
  writeln('                     1. Learn Lessons');
  writeln;
  writeln('                     2. Do Quiz');
  writeln;
  writeln('                     3. Play Games');
  writeln;
  writeln('                     4. Change Password');
  writeln;
  writeln('                     5. Logout');
  writeln;
  writeln('                ------------------------------------');
  writeln;
  write('                 Enter your choice: ');
  textcolor(14);
  readln(Choice);
  writeln;
  case choice of
    1 : lesson_menu;
    2 : do_quiz;
    3 : play_game;
```

```
    4 : change_password(user_index);
    5 : logout_confirm
  end;
end;

{Creating account}
procedure create_acc;
var
  loginid, username, password1, password2 : string;
  name_OK, id_OK, pw_OK : boolean;
  i : integer;
begin
  name_OK := false;
  repeat
  clrscr;
  writeln;
  textcolor(14);
  writeln('       Your Name should be at most 25 characters');
  writeln('       Your Name cannot be changed after input.');
  textcolor(7);
  writeln('       Please enter your Name: ');
  textcolor(15);
  write('       ');
  readln(username);
  if length(username) > 25 then
     begin
       writeln;
       textcolor(12);
       writeln('       The length of your name can not be more than 25
characters!');
       write('       Press <Enter> to retry. ');
       readln
     end
  else
     name_OK := true
  until name_OK;

  id_OK := false;
```

```
repeat
  clrscr;
  writeln;
  textcolor(10);
  writeln('       Welcome ', username, '!');
  writeln;
  textcolor(14);
  write('       Please enter your UserID (4 characters) : ');
  textcolor(15);
  readln(loginid);
  if length(loginid) <> 4 then
    begin
      writeln;
      textcolor(12);
      writeln('       The length of UserID must be 4 characters!');
      write('       Press <Enter> to retry. ');
      readln
    end
  else
    begin
      id_OK := true;
      for i := 1 to num_user do
        if loginid = userid[i] then
          begin
            writeln;
            textcolor(12);
            writeln('       The UserID has been used!');
            write('       Press <Enter> to retry. ');
            readln;
            id_OK := false
          end
    end
until id_OK;

pw_OK := false;
repeat
  clrscr;
  writeln;
```

```
textcolor(10);
writeln('       Welcome ', username, '!');
writeln;
writeln('       Your UserID is ', loginid);
writeln;
textcolor(14);
write('       Please enter your password (4 characters) : ');
textcolor(15);
password1 := GetPword;
if length(password1) <> 4 then
  begin
    writeln;
    textcolor(12);
    writeln('       The length of password must be 4 characters!');
    write('       Press <Enter> to retry. ');
    readln
  end
    else
      begin
        writeln;
        textcolor(14);
        write('       Please enter your password again    : ');
        textcolor(15);
        password2 := GetPword;
        if password1 <> password2 then
          begin
            writeln;
            textcolor(12);
            writeln('       The passwords do not match!');
            write('       Press <Enter> to retry. ');
            readln
          end
        else
          begin
            num_user := num_user + 1;
            username := username + '                      ';
            username := copy(username, 1, 25);
```

```
            userid[num_user] := loginid;
            userpw[num_user] := password1;
            name[num_user] := username;
            store_user_info;
            pw_OK := true;
            writeln;
            textcolor(14);
            writeln('       User account created successfully.');
            write('       Press <Enter> to return. ');
            readln
          end
        end
  until pw_OK
end;

{Login function}
procedure login(var user_index : integer);
var
  loginid, password : string;
  found : boolean;
  i : integer;
begin
  clrscr;
  writeln;
    Textcolor(10);
    writeln('                          Welcome to');
    Textcolor(11);
    writeln('                          __                              ');
    writeln('                ___  / /   ___  ____ _____  ___  _____');
    writeln('               / _ \/ /   / _ \/ __ `/ __ \/ _ \/ ___/');
    writeln('              /  __/ /___/  __/ /_/ / / / /  __/ /    ');
    writeln('              \___/_____/\___/\__,_/_/ /_/\___/_/     ');
    writeln;
    Textcolor(10);
    writeln('                          Please login:');
    writeln;
    writeln('               ------------------------------------');
  writeln;
```

```
  write('                      UserID : ');
  readln(loginid);
  writeln;
  write('                    Password : ');
  password := GetPword;
  writeln;
  writeln;
  found := false;
  i := 0;
  while (i < num_user) and (not found) do
    begin
      i := i + 1;
      if (loginid = userid[i]) and (password = userpw[i]) then
        begin
          found := true;
          user_index := i
        end
    end;
  if not found then
    begin
      user_index := 0;
      textcolor(12);
      writeln(''':20,'> > > Invalid UserID or Password!');
      write(''':20,'> > > Press <Enter> to return.');
      textcolor(15);
      readln
    end
  else
    main_menu(user_index)
end;

{login page}
procedure intro;
var
  choice : integer;
begin
    clrscr;
    writeln;
```

```
    Textcolor(10);
    writeln('                          Welcome to');
    Textcolor(11);
    writeln('                        __                              ');
    writeln('                 ___  / /   ___  ____ _____  ___  _____');
    writeln('                / _ \/ /   / _ \ __ `/ __ \/ _ \ ___/');
    writeln('               /  __/ /___/  __/ /_/ / / / /  __/ /    ');
    writeln('               \___/_____/\___/\__,_/_/ /_/\___/_/     ');
    writeln;
    Textcolor(10);
    writeln('                          Please choose:');
    writeln;
    writeln('                ------------------------------------');
    writeln;
    writeln('                     1. Login');
    writeln;
    writeln('                     2. Register');
    writeln;
    writeln('                ------------------------------------');
    writeln;
    write('                     Enter your choice: ');
    textcolor(14);
    readln(Choice);
    writeln;
    case choice of
      1 : login(user_index);
      2 : create_acc;
    end;
end;

{main body}
begin
  read_user_info;
  read_notes;
  status := 1;
  repeat
    case status of
      1 : intro;
```

```
      2 : main_menu(user_index);
      3 : lesson_menu
    end;
  until status = 0;
  textcolor(10);
  writeln;
  writeln;
  writeln;
  writeln('                    Thanks for using eLearner!');
  readln
end.
```

Appendix 2 - Working schedule

Date	Event
March 2015	Choice of topic
April 2015	Background research + Define the objectives
Early May 2015	Search on Internet for details of making an educational software
Late May 2015	Search on Internet for information about ICT related knowledge
June – July 2015	Design of solution
August – October 2015	Implementation
Early September 2015	Ask teachers for advice
October 2015	Finish most of the program with temporary file
Early November 2015	Make all the resources files as a sample
November – December 2015	Testing
Early January 2016	Conclusion & Discussion + Final Report
Mid-January 2016	Final check of the program and the report
After all things done	Hand in the report and program by making them into a zip file