

Hong Kong Diploma of Secondary Education

Examination **xxxx**

Information and Communication Technology

(Coursework)

Option D: Software Development

Title: Educational Software Program

Candidate ID No:

Content

1

Chapter 1 Introduction	3
1.1 Introduction	3
1.2 Objectives.....	4
1.3 Project Plan.....	7
Chapter 2 Analysis	8
2.1 Leaner Regulations	8
2.2 Data Collection	9
2.3 Input, Output and Process	10
2.4 Choice of IT Tools.....	13
2.5 Conclusion of Study	13
Chapter 3 Design of Solution	14
3.1 Brief Description	14
3.2 Refinement of Problem.....	15
3.3 Input Data File Formats.....	19
3.4 Output Files/Data Format.....	28
Chapter 4 Implementation	39
4.1 Brief Description	39
4.2 Data Structures.....	39
4.3 Procedures in the Program	51
4.4 Program Coding	63
4.5 Program Execution	64

Chapter 5 Testing & Evaluation.....	82
5.1 Brief Description	82
5.2 Testing and Evaluation Plan.....	83
5.3 Internal Testing	84
5.4 Self-Evaluation.....	104
5.5 External Testing and Evaluation	105
5.5.1 Evaluation	114
Chapter 6 Conclusion & Discussion.....	115
6.1 Pros and cons of my Program	115
6.2 Future Improvement.....	117
6.3 Self-Reflection.....	119
Chapter 7 Reference and Acknowledgement	121
Appendices	122
Appendix 1 – Program Code (after Testing & Evaluation)	122
Appendix 2 - Working schedule	160
Appendix 3 – User Manual	161

Chapter 1 Introduction

1.1 Introduction

Situation

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

Background Information

With no doubt, many students have a computer at home and they make use of it to study.

With computers, unlike the past by handling learning resources by paper, resources could be analyzed more accurately and give results in the fastest way. Also, in the century which is promoting **Green- Living Style**, the use of computers **reduces the wastes of papers** and the time of printing. The students can also learn by playing games by computers. Thus, many teachers are using online systems such as Eclass System, Facebook or DropBox etc. to provide worksheets, exercises and quizzes for the students. Information system is widely used in companies, schools and families. These are examples of computer systems that handling information and the universality of the use of computer system.

The information system could be applied to an educational software program.

In this project, the program will be developed as a kind of application that uses in different computers at different locations, and with different users. The educational system is developed to provide the information of **Information & Communication Technology** in the secondary school syllabus to **assist students' learning**. The functions such as a series of lessons, quizzes and educational games, etc. will be provided in the program.

The educational software program will be named "**ICT_Learner**"



1.2 Objectives

Problems

As mentioned, this educational software program aims at providing different functions such as a series of lessons, quizzes, educational game, etc. to teachers and students in order to assist the students learning.

Targeted users

As the educational software program is used to assist students learning. Two types of users will be involved:

1. Teachers, who are teaching Information & Communication Technology, responsible to assist students learning through the program.
2. Senior form students, who are taking Information & Communication Technology responsible to learn through the program.

Users' responsibility

1. Teachers
 - To monitor the students learning progress
 - To provide resources to students for learning
 - Give comments or supports to students
 - To edit the learning resources
2. Students
 - To learn and improve their performances.
 - Complete the quizzes or homework.
 - Give comments or their own questions to teachers
 - Learn Information & Communication Technology individually and joyfully.

Login

Login process in this system is required in order to identify their identity and authorizing appropriate access right to the user.

Each teacher has their own login account to the system. List of students will be accessed by teachers to understand the learning process of students.

Each student has their own login account to the system. List of lessons, quizzes, games, etc. will be provided.

Functions

In the educational software program, there are mainly two types of users- teachers and students. Different kinds of functions will be provided to assist students learning.

For Teachers:

Firstly, they are responsible for Learning Resources Panel. In the program, ready-made files of a series of lessons, quizzes and educational games are provided to use. Teachers can edit the resources in order to update them, they may edit them directly in the files. Teachers may create new file in the folders that provided by the program. Learning Resources Panel contains subject lessons, quizzes and educational games, etc.

Secondly, they are responsible for the Students List Panel. It is an important panel that consists of students' information and the results of quizzes and games. This panel is mainly used to monitor the learning progress of the students. Teachers can also look for a specific student for his results by the function of searching the student from the list.

Thirdly, there will be a Comment Panel for the teachers to give simple comments or supports to their students. Teachers may also read students comments from the panel of each student.

For Students:

Firstly, they are responsible to use the learning resources from the Learning Resources Panel. They couldn't edit any files in the panel. However, they can read the series of lessons, do quizzes and play games. They can choose the topic or chapter they want to read. They can learn **Information & Communication Technology** using this panel. They can also test themselves through quizzes and enjoy the games.

Secondary, they can read their own students' record from the Student Record Panel. The record and scores of the quizzes will be stored in the Student Record Panel.

Thirdly, from the Comment Panel, students can read the comments from their teachers. They can give simple comments or questions to their teachers.

Walkthrough for the functions

- Functions for teachers
 - ✓ Allow teachers to create & edit learning resources such as contents of lessons, quizzes, and games. Etc.
 - ✓ Allow teachers to monitor students records
 - ✓ Allow teachers to search the students records
 - ✓ Allow teachers to give comments
- Functions for students
 - ✓ Choose and read a series of lessons
 - ✓ Choose and play games
 - ✓ Do exercises
 - ✓ Do quizzes
 - ✓ View their quizzes' results
 - ✓ Give comments, suggestions and questions to teachers
- Types of quizzes/worksheets
 - ✓ Multiple choices
 - ✓ TRUE OR FALSE
 - ✓ Fill in the blanks
 - ✓ Matching
- Types of games
 - ✓ Vocabulary Guessing Game(different levels)
 - ✓ Matching
 - ✓ Fishing
 - ✓ Car Racing Competition

Details of the above proposed function will be described in next chapter.



1.3 Project Plan

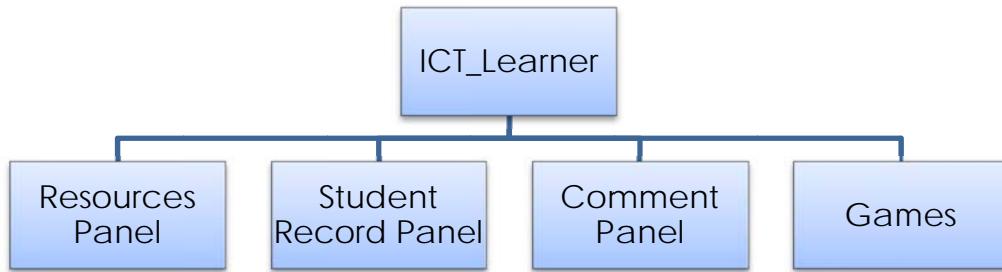
- I will solve the problem according to the following procedure:
 1. Analysis:
 - Define the learner regulations, such as rules, login account etc.
 - Identify the input, processing, and output of the program
 - Based on the proposed functions, study alternative ways of solving the problem.
 - Justification of the choice of appropriate IT tools for solving the problem.
 2. Design
 - Design the overall structure of the program
 - Design different parts of program
 - Design the formats of the input files
 - Design the outline of files
 - Design the layout of the analysis report
 3. Implementation
 - Decide the data structures that will be used in the program.
 - Design the algorithms according to the proposed functions, the learner regulations, etc.
 - Write different parts of program and test them
 - Put them into a main program
 - Construct the program
 4. Testing and Evaluation
 - Design the test plan for the program.
 - Perform testing and evaluation on the program according to the proposed test plan.
 - Debug the program
 - Correct any mistakes
 - Use Setup Creator to make an application.
 5. Conclusion and Discussion
 - Summary of the project
 - Conclusion on what have been studied/learned
 - Discussion on the favorable features and shortcomings of the program
 - Suggestion on further development/improvements on the program.



Chapter 2 Analysis

2.1 Learner Regulations

The whole system can be simply divided as four components:



Resources Panel is used for reading notes, doing exercises and quizzes.

Student Record Panel stores and show records.

Comment Panel act as a platform for students and teachers to communicate.

Feasibility Study

Technical Feasibility

For the feasibility on the equipment, what the system required are simply a computer, a keyboard. The monitor should have a resolution of 1024 x 768 to have better performance. These are the tools developed for a long history which are not any new technology.

For the feasibility on the software technology, what the system required are simply a Microsoft OS (Windows) and the ICT_Learner to handle the input, process and output functions.

Economic Feasibility

As this system can reduces a lot of human work – distributing notes, exercises and quiz, marking quiz and exercises, etc. These human work processes involve a lot of human cost. However, through iLeaner, with its low building cost or running cost, also it reduces the use of papers, it is beneficial to both schools and environment.

Assumptions and Information

ICT_Learner regulations and assumptions on the problem:

- Each user must have an account and belongs to one of the ICT classes
- Maximum number of students each class is 30
- Assume that 1 form has only one class
- Maximum teacher number is 2 for each form
- Using the same set of questions for all students
- 5 questions at most in a question set?
- Maximum 9 cards for matching game
- Questions Type
 - o Filling the blank
 - o True or False
 - o Multiple Choice
 - o Matching(matching game)
- There are 2 to 4 choices in one MC question
- Choices: A, B, C, D
 - 1, 2, 3, 4, 5...
 - T, F
- There are different levels of questions
- All questions carry equal mark
- Students can do quizzes and exercises many times
- Only the first 3 quizzes will be count and the marks will only be recorded once
- Students should not use rude words in the comment panel
- Exercises marks are not recorded
- Awards will be decided by teachers
- For games, students can record their score by screen shot and give teachers for award.

2.2 Data Collection

- Data Collection Method
 - The Data (learning resource) will be collected on Internet OR type by teacher.
 - *The Account and Password will give to the students through E-mail and paper by teacher*
 - Comment Panel is ready-made
 - Account is same as student name
 - Password is their registration number
 - Record of all quiz score is -1
- Text files to be collected
 - 3 text files containing all the students accounts and passwords
 - 3 text files containing all the students record
 - text files for notes, exercise, quizzes , game resources and the lists of resources

2.3 Input, Output and Process

Functions & Construction of Sub-programs

Data Management

Data management is used to manage the resources and data in the system. Normally, what data involved in a learning system would be information of accounts, students, teachers, notes, exercises, quizzes, game resources. A Panel can be considered as a module which provides functions.

Login Panel

- Used to store accounts' information
- Searching
- Identify the user is teacher or student
- Required Information:
 1. Student's English Name
 2. Password
 3. Class Form

Resources Panel

- Note Panel
 - Store Notes
 - Read Notes
- Exercise Panel
 - Store Exercises
 - Do Exercises
- Quiz Panel
 - 1. Store Quizzes
 - 2. Do quiz
 - 3. Update the quiz score

Student Record Panel

- Student
 - Search and show their own records
- Teacher
 - List all record in that form
 - Required Information
 1. Class
 2. Name(For student)

Practical Function

Practical function is a set of functions that will be used during the program.

#After the Welcome Page pop up and user type to continue, the following panel will be used.

The location of the main folder (ICT_Learner) of program will be gotten by a procedure (discuss in next chapters)

Login Panel

- Used for identification
- Required Information
- Input:
 1. Status
 2. Form
 3. Account(Student English Name)
 4. Password
- Storage:



Account (name)

Resources Panel

Main Panel for learning:

- ❖ Note Panel
 - Read Notes
- ❖ Exercise Panel
 - Do Exercises
- ❖ Quiz Panel
 - Do quiz
 - Update the quiz score

Required Information

- ❖ Input
 1. Choice for chapters
 2. Choice for answering question
- ❖ Storage:
 1. Notes
 2. Exercise
 3. Quiz
 4. Quiz Score

Student Record Panel

- Show Students Record
- Required Information

▪ Storage:

1. Status
2. Name
3. Quiz Scores



Comment Panel

- Provide a place for students and teachers to communicate about ICT
- Required Information

▪ Input

1. Comment
2. Choice for quit

▪ Storage:

1. Comments File
2. Name
3. Quiz Scores

Game Panel

- Provide a place for students and teachers to play games about or not about ICT
Students learn ICT through games
- Required Information

▪ Input

Choice For Game

▪ Storage:

Games Resources

The Data Flows will be detailed In the next chapter!

2.4 Choice of IT Tools

As aforementioned, this application would be launch and updated in different position in a school. Thus, connection between computers at schools is needed. If there are different windows, I have to develop a program that can be executed in different windows. For this, we have 3 choices – Dev-Pascal, Visual C++ and Visual Basic.

In order to reduce the loading of computer and make the program available for most of the students, I decided to use Dev-Pascal. The Dev-Pascal mainly contains two modes text-mode and graphic-mode.

Dev-Pascal can make a program running DOS and using only a little resource of CPU. Students and teachers can perform other task while using ICT_Learner. The system is performed by simple English which is easily understood by most of the students.

The following Program will be developed under a 1024*768 resolution computer. To have a better performance, having at least 1024*768 resolution monitor is recommended. 1024*768 resolution is very common these days, so I choose to develop the program in this condition.

2.5 Conclusion of Study

Conclusion

This system will provide two main functions, which are data management and practical functions. Data management, or data warehouse, is used to store data such as students, records, accounts and learning resources while use, including do exercises, quizzes and comment panel will be functions used while learning. The program aims at assist students' learning and increases their interests towards ICT.



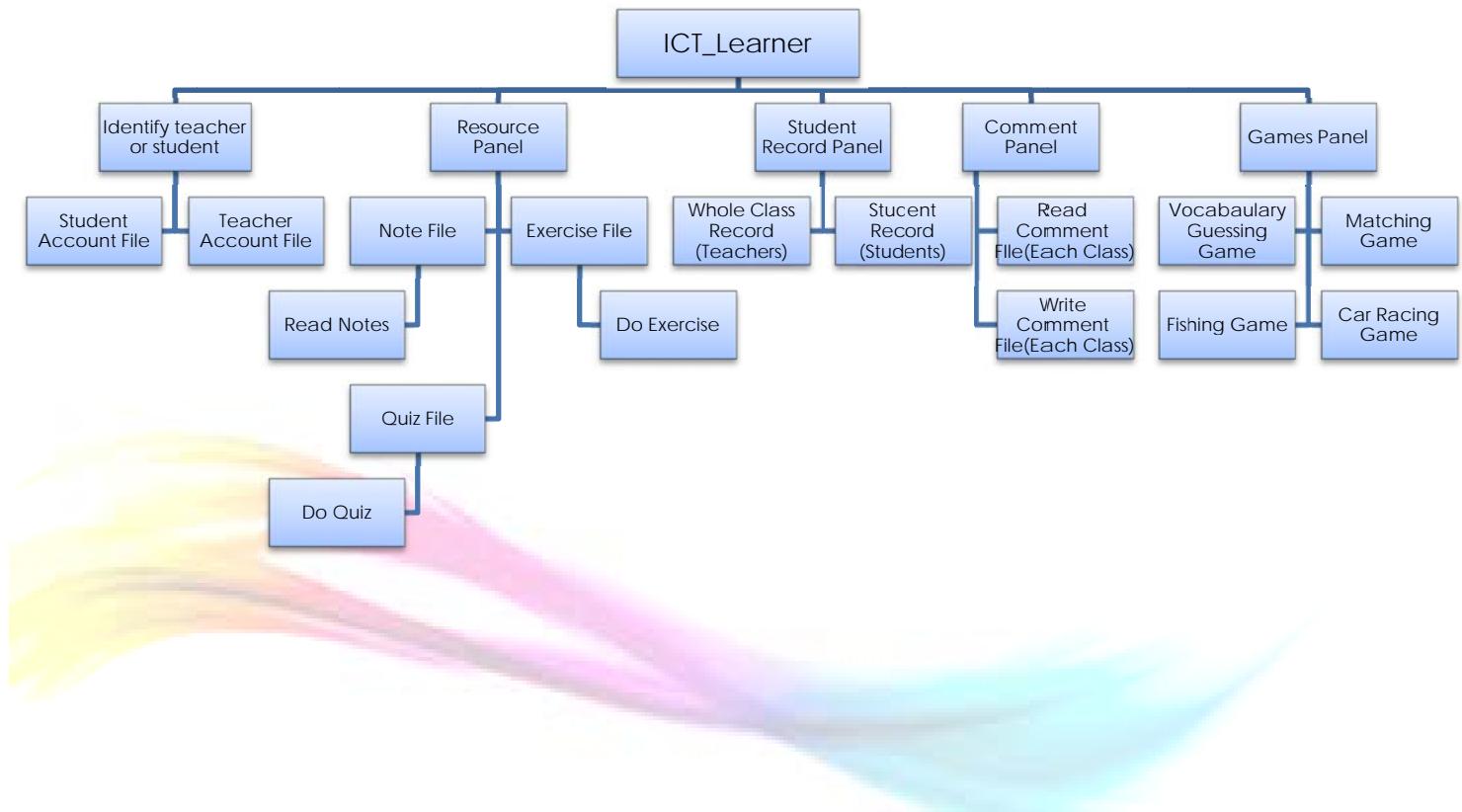
Chapter 3 Design of Solution

3.1 Brief Description

- In this Chapter, I will design the program based on the functions I proposed in Chapter 1, and the input, process and output I studied in Chapter 2
 - I will design:
 1. the overall structure of the program by refining the problem
 2. the formats of the data files for storing
 - Login Accounts and Password
 - Notes
 - Exercises
 - Quizzes
 - Students Records
 - Games Resources
 3. the data flow of the program
 4. The Input samples
 5. The Output Samples



3.2 Refinement of Problem



Here is the big mind map showing what function will be developed for this program.

Data Flow:

#After the Welcome Page pop up and user type to continue, the following panel will be used.

#The location of main folder (ICT_Learner) in the hard disk will be found by a short procedure.

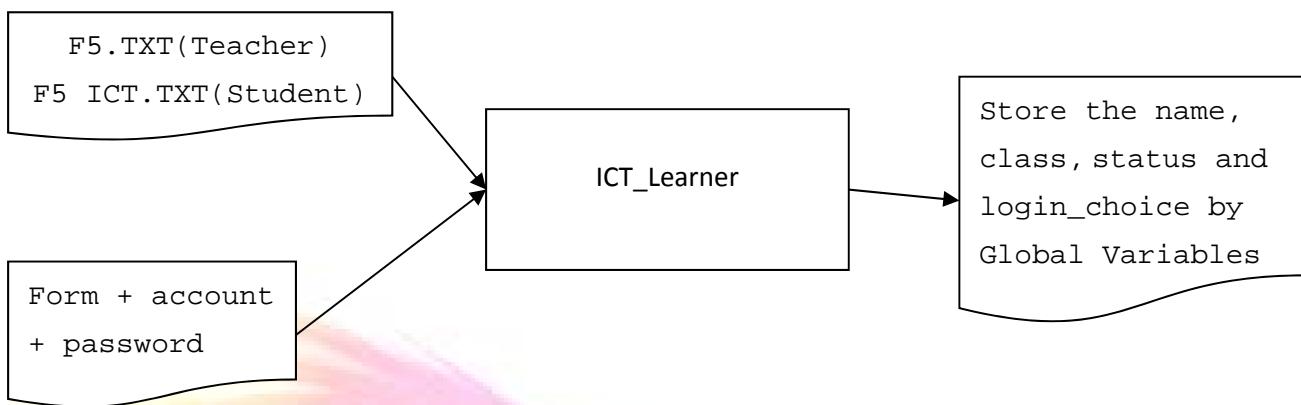
Assumption for example: The File used is assumed that the student or teacher is F5

*****All files can be edited directly through Notepad/WordPad by teachers as long as they follow the formats. (Edit the *.txt file)

Login Process

*The Account and Password will give to the students through E-mail and paper by teachers.

Open Student_Accounts or Teacher_Accounts Folder (identify by status = S/T)



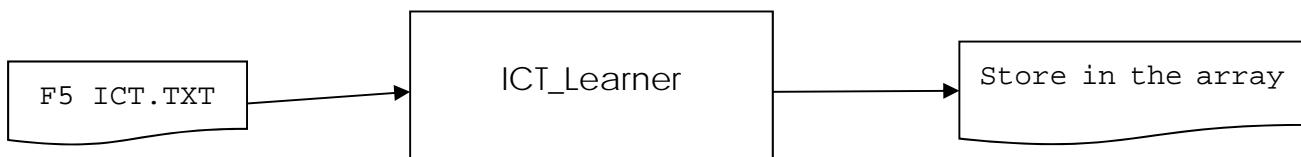
Check if the student is using his own account and store all his information for further use.

Student Record Panel

Store Record

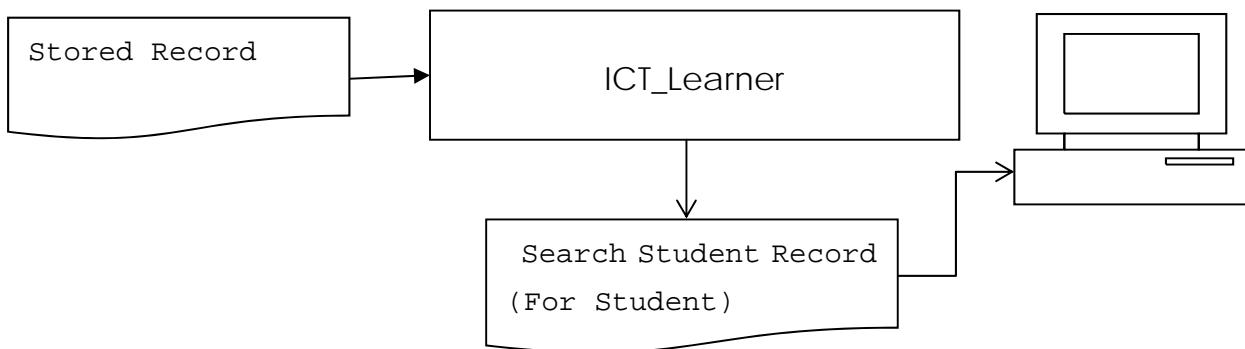
After the login process, the records are stored in a record type array

Open Records Folder



Open the record file and store in the array variable after login for further use.

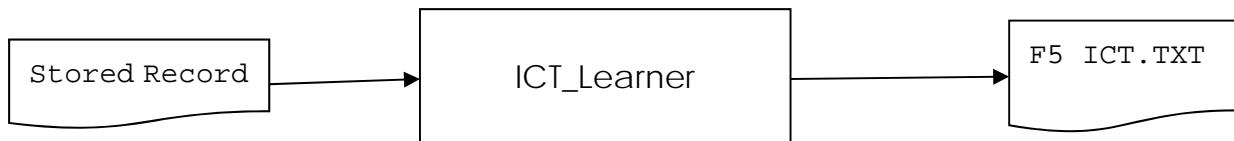
Show Record



Used to show record of a student or the record of whole class

Update Record

After the users choose to quit:



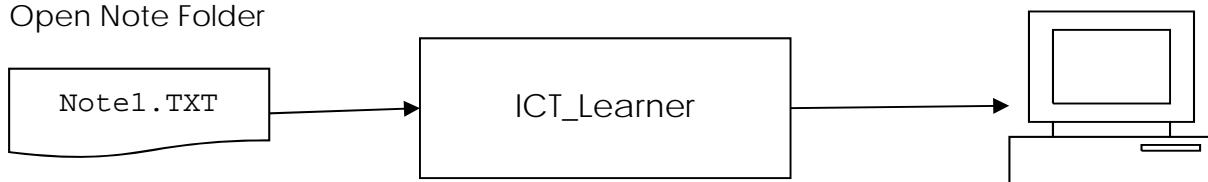
Used to Update the record file after the user quit

Resources Panel

Assume First resource of each part:

Read Notes

Open Note Folder



This part uses to read a note after user choose one

Do Exercises

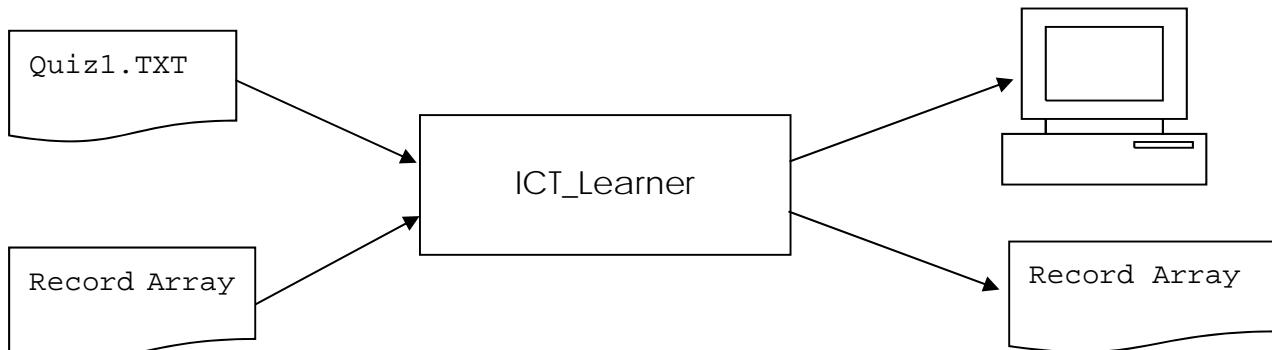
Open Exercises Folder



This part uses to open an exercise and let users do after user choose one

Do Quizzes

Open Quiz Folder

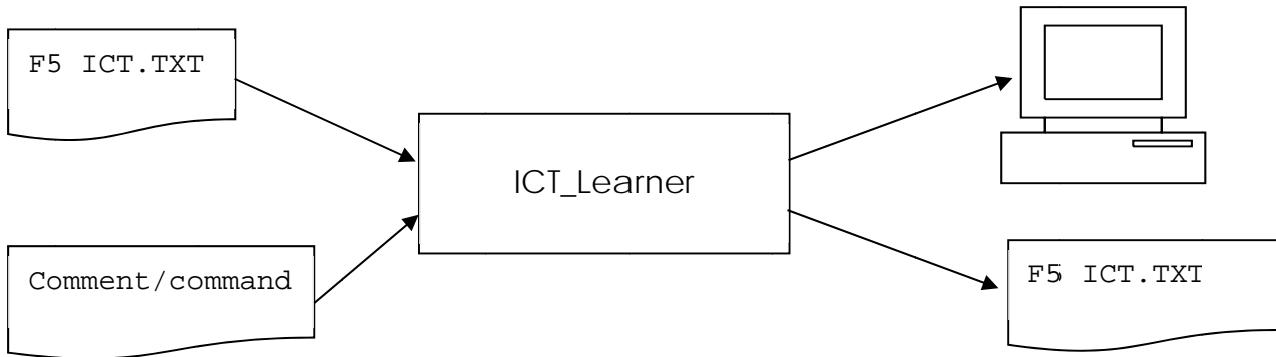


This part uses to open a quiz and let users do after user choose one

Comment Panel

18

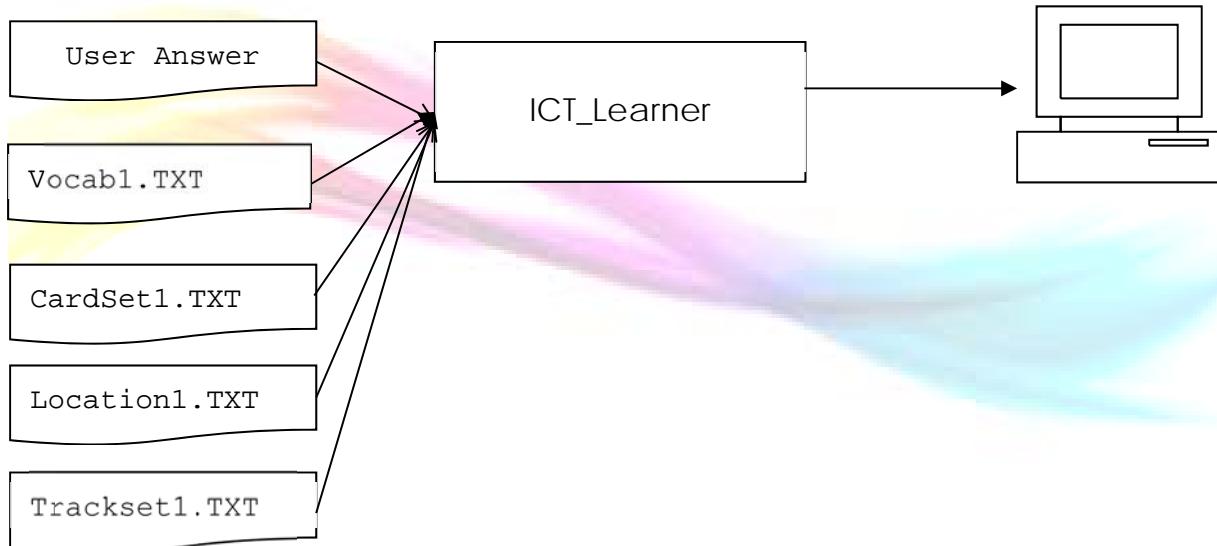
Open comment folder



These Panel acts as a platform for communicate.

Games Panel

Open game resources folder



This part is the sub-program for playing educational games



3.3 Input Data File Formats

- File storing the Students' and Teachers' Account and Password:

- 3 files for students and 3 files for teachers
- File name: 'F ' + form inputted by user + ' ICT' + '.txt' (Students)
- File name: 'F ' + form inputted by user + '.txt' (Teacher)
- File type: plain text file
- Data stored:

The data file stores the account and record of each student, which includes the following information:

- o Account (string)
- o Password (string)

- File structure:

Each line of the data file stores the account and password of user with the following format:

- *Short Brief: The teacher may take the student name as account and student ID as password. (Inform the students by E-mail or papers)

e.g.

Account (String)	Password (String)
Thomas Chung	s503

- Sample file:

F5.ICT.TXT (Student)

```
Account Password
Thomas Chung s503
```

F5.TXT (Teacher)

```
Account Password
chukf 123456
```

Sample input:

What is your form? (e.g.4): 5

What is your name?: Thomas Chung

Password: s503

2. File storing student record of quizzes 1-3

- 3 files
- File name: Class + '.txt' (e.g. F5 ICT.txt)
- type: plain text file
- Data stored (for one exercise/quiz only):

The data file stores the score record for quiz 1-3 of students
- File structure:
 - ◆ Each Line contain the name of the student
 - ◆ The next line is his scores
 - ◆ Name = account
 - ◆ The score -1 means the student haven't done the quiz yet.
- *****Remark: Teacher SHOULD set the students records to -1 (NOT DONE) for new class or new quiz. These files are needed to be edited before using.
- Sample file:

F5 ICT.TXT

Name
Quiz1 Quiz2 Quiz3
Thomas Chung
-1 -1 -1

The file data will first store in arrays.

After that, the updated arrays data will overwrite the record file to update the record.

The record shows according to:

Status

Name

Class

3. File storing Notes

- 3 -5 files for notes
- File name: Note + number inputted by user+'.txt' (e.g. Note1.TXT)
- type: plain text file
- Data stored (for one note):
The data file stores notes of one chapter or some chapters
- File structure:
 - ◆ Chapter Name
 - ◆ Contents
- Short brief: Teachers can edit it by notepad.
- Sample file:
- Note1.TXT

```
1.1 A Sample of notes  
It is just a sample of notes.
```

Sample Input:

Your Choice: 1



4. File storing Exercises and Quiz:

- 3 files for exercise. 3 files for quiz at most.
- File name: Ex + number inputted by user+'.txt' (e.g. Ex1.TXT)
- File name: Quiz + number inputted by user+'.txt' (e.g. Quiz1.TXT)
- type: plain text file
- Data stored (for one exercise/quiz only):
 - The data file stores one set of exercise/quiz questions and answer
- File structure:
 - First Line is number of questions in exercise/quiz.
 - Start from second line:
 - Question
 - Choice of question/information of answering
 - Answer of question
- Short brief: Teachers can edit it by notepad.
- Sample file:
- Ex1/Quiz1.TXT

```
3
Question 1
Choice
Answer
Question 2
Choice
Answer
Question 3
Choice
Answer
```

Sample Input:

Question 1
 How are you?
 1)I'm fine 2)Do I know you?
 Answer ? 1

5. File storing Comment

- 1 file for 1 class
- File name: Class+'.txt' (e.g. F5 ICT.TXT)
- type: plain text file
- Data stored (for one exercise/quiz only):
The data file stores whole class comments
- File structure:
Lines of comments
- Sample file:

F5 ICT.TXT

```
admin:you can type your comment here
```

Sample Input:

What comment you want to add?(Type <Q> to Exit)

Hello Mr. Chu I want to ask you about the deadline of homework



6. File storing Game Resources

6.1 Vocabulary Guessing Game

3 files storing the vocabularies

File name: Vocab+ number inputted by user+'.txt' (e.g. Vocab1.TXT)

- type: plain text file
- Data stored (for one set):
Vocabularies about ICT or common vocabularies
- File structure:
One vocabulary for one line
- Sample file:

Vocab1.TXT

```
computer
device
implement
```

Sample Input:

The secret word to be guessed is *****

No. of Attempts remaining = 12

Please input the letter you guess: d

6.2 Matching Game

3 files storing the cards and answer match

File name: CardSet +number inputted by user+'.txt' (e.g. Vocab1.TXT)

- type: plain text file
- Data stored (for one set):
number of cards, the correct match and cards contents
- File structure:
- First line : number of cards
- Second to number of card /2 line : the correct match
- After the correct match:
30 * 8 characters cards include the card outline

- Sample file:

CardSet1.TXT

```
2
1-2
1*****
*****
*****
*****
*****
*****
*****
*****
2*****
*****
```

Sample Input:

The 1st card you want to flip: 1

The 2nd card you want to flip: 2

6.3 Fishing Game

3 files storing the cards and answer match

File name: Location+ number inputted by user +' .txt' (e.g. Location1.TXT)

- type: plain text file
- Data stored (for one set):
number of fishes (questions), choice, answer
- File structure:
- First Line is number of fish(question) in the location(file)
- Start from second line:
 - Fish (Question)
 - Bait (Choice of question/information of answering)
 - Answer of question
- Sample file:

Location1.TXT

```
3
Fish 1 (Question 1)
Bait(Choice)
Answer
Fish 2 (Question 2)
Bait(Choice)
Answer
Fish 3 (Question 3)
Bait(Choice)
Answer
```

Sample Input:

Fish 1

How are you?

1)I'm fine 2)Do I know you?

Answer ?

1

6.4 Car Racing game

3 files storing the cards and answer match

File name: Trackset+ number inputted by user +'.txt' (e.g. Trackset1.TXT)

- type: plain text file
- Data stored (for one set):
number of laps (questions), choice, answer
- File structure:
 - First Line is number of laps(question) to race
 - Start from second line:
 - Lap(Question)
 - Move(Choice of question/information of answering)
 - Answer of question
- Sample file:

Trackset1.TXT

```
1
HELLO World?
1)HI 2)Sorry I Don't know you
1
```

Sample Input:

Lap1

HELLO World?

1)HI 2)Sorry I Don't know you

Answer ? 1

3.4 Output Files/Data Format

Login Progress

For this part, after the user type in the form, account and password it will directly show the Main Menu and a place for choosing.

1. Sample output on screen

The Main Menu

Choice area

Sample layout:

```
Main Menu
1.Learning Resources Panel
2.Students Record Panel
3.Comment Panel
4.Games
5.Quit
Your Choice:
```

Student Record Panel

For this panel it divided to 3 parts – store records, show records and update records

Store record

After the user login, the record will be stored a record type array which has been discussed in the previous section.

Show record

When the user chooses to login in the record panel, by identifying the user's status (teachers or students), their own record or whole class record will be shown. (Discussed in previous sections)

1. Sample output on screen

Name Scores

Sample layout:

Name	Quiz1	Quiz2	Quiz3
Thomas Chung	-1	-1	-1

Update record

When the user chooses to quit the program, write all arrays of new records into the file
 (Discussed in previous sections)

1. Sample output in file:

Name

Scores

Sample layout:

```
Name
Quiz1 Quiz2 Quiz3
Thomas Chung
-1 -1 -1
```

Resources Panel

This panel also has 3 sub-programs –read note, do exercise and do quiz.

The output below is the sub-program after choosing topic/chapter. Therefore, the list of resource is not shown. (There are files storing the list of resources).

Sample output of choosing list: (use quiz as example) (Same in notes, exercise, etc.)

```
Quiz Menu
1:Quiz 1
2:Quiz 2
3:Quiz 3
4:Resources Menu
Your Choice:
```

The choice number for “getting back to menu” bases on how many resources are there.

Read Note

When the user chooses a note, the program search the note file by the note number (Discussed in previous sections)

1. Sample output on screen

Notes....

.....

.....

A choice area to quit

Sample layout:

```
Chapter 1: What is a computer?
.....  

.....  

.....  

Press Q to back to Notes List
Your Choice:
```

Do Exercises and Do Quizzes

These 2 parts are very similar! Almost the same.

When user chooses a note, the program searches the exercise or quiz by its number (Discussed in previous sections)

1. Sample output on screen:

A simple explanation

Question number

Question

Choices

Answer space

Check answer

Results

A choice area to quit

Sample layout (Correct answer) (Exercise):

```
There are 1 question(s) in this exercise
```

Question 1

Are you here?

A)Yes B)No

Answer?A

Right!

You have 1 mark(s)

Press Q to back to Exercises List

Your Choice:

Sample layout (Wrong answer) (Exercise):

```
There are 1 question(s) in this exercise
```

Question 1

Are you here?

A)Yes B)No

Wrong!

Answer is : A

You have 0 mark(s)

Press Q to back to Exercises List

Your Choice:

Sample layout (Correct answer) (Quiz):

There are 1 question(s) in this quiz

Question 1

Are you here?

A)Yes B)No

Answer ?A

Right!

You have 1 mark(s)

Answer will be given by teacher.

Press Q to back to Exercises List

Your Choice:

Sample layout (Wrong answer) (Quiz):

There are 1 question(s) in this exercise

Question 1

Are you here?

A)Yes B)No

Wrong!

You have 0 mark(s)

Answer will be given by teacher.

Press Q to back to Exercises List

Your Choice:

Quiz scores will be stored into the arrays.

Comment Panel

When get into the comment panel, the program search the comment file of his class. User can read and write comments. (Discussed in previous sections)

1. Sample output on screen

Comments

Area of typing comment

Sample layout (screen):

admin:you can type your comment type

What comment you want to add?(Type <Q> to Exit)Are you here?

OK hellooooo

What comment you want to add?(Type <Q> to Exit)Are you here?

Sample layout (comment file):

admin:you can type your comment here

F5 ICT Thomas Chung: OK hellooooo

Games Panel

There are 4 games in this panel.

Vocabulary Guessing Game

In this game, the user first chooses a chapter /set he wants to guess a word from. The program randomly chooses a word from that file and set it as a secret word (word to be guess). (Discussed in previous sections)

1. Sample output on screen

Secret word

Time remaining for guessing

Area of guessing

Sample layout (correct guess and win):

The secret word to be guessed is *****

No. of Attempts remaining = 14

Please input the letter you guess: t

t***t***

No. of Attempts remaining = 13

Please input the letter you guess:

.

.

No. of Attempts remaining = 8

You win

Press Q to back to Games_Menu

Your Choice:

Sample layout (wrong guess and lost):

The secret word to be guessed is *****

No. of Attempts remaining = 14

Please input the letter you guess: f

No. of Attempts remaining = 13

Please input the letter you guess:

.

Please input the letter you guess: g

*****g

No. of Attempts remaining = 0

You lose

Press Q to back to Games_Menu

Your Choice:

Matching Game

In this game, the user first chooses a set of he wants to play. The content of card will be read from file and store the information.

The cards are covered, if the user got correct pair, the cards will be flipped and remain there. If the user is wrong, the cards flip back and cover.

1. Sample output on screen

Cards

First guess

Second

Tell wrong or not.

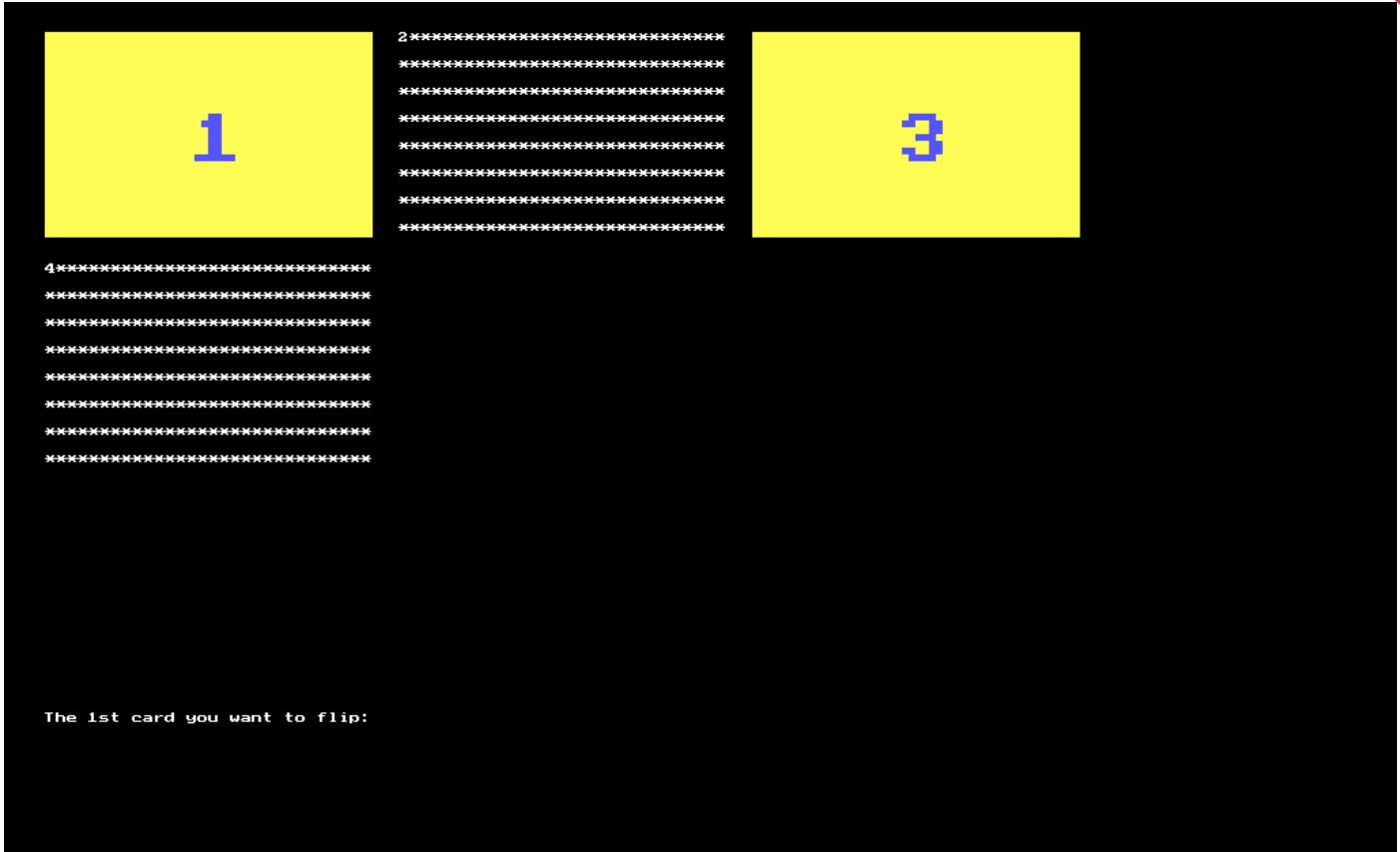
Sample output on screen:

Correct pair:

```
1*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
2*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
3*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
*****  
The 1st card you want to flip: 2  
The 2nd card you want to flip: 4  
You Have The Correct Match!  
You Tried 2 Time(s)  
You Have Got 1 Pair(s)
```

Correct pair after some time:

34

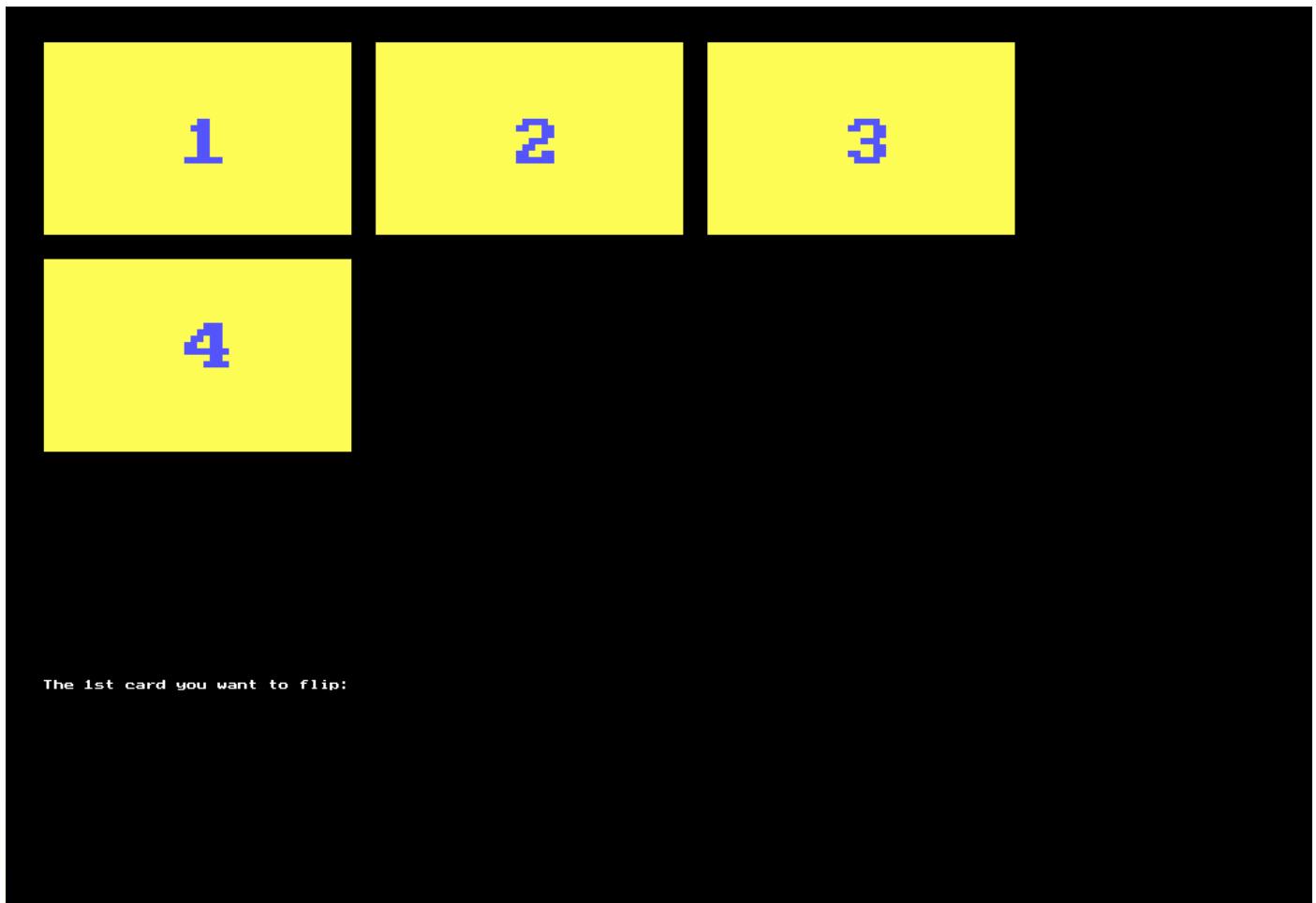


Wrong Pair:



Wrong Pair after some time:

35



The Screen will show 'You Have Got All Pairs Back To Main Menu In 5 Seconds' After the game is finished.

Fishing Game

Once the user chooses a location, the program will read all the question of the file. This sub-program is quite similar to do quiz and do exercise

1. Sample output on screen

A random fish (from fish file)

Fish Number (Question number)

Fish (Question)

Bait (Choices)

Answer space

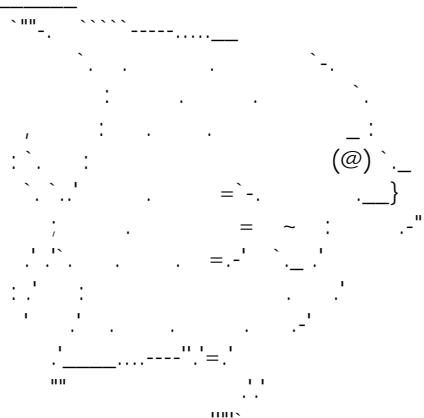
Check answer

Results

A choice area to quit

Sample layout:

36



Fish 1

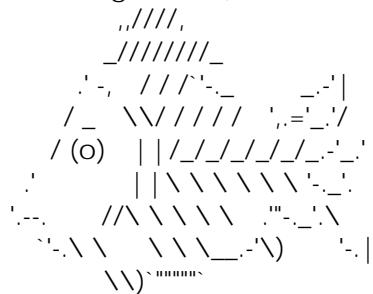
HEIO HERE JUST TESTING

1)Quit 2)Ha

Answer?1

You Caught The Fish!

{After clearing Screen}



Fish 2

HEIO HERE JUST TESTING2

1)Quit 2)Ha

Answer?2

You Missed The Fish!

The Fish Wanted Bait 1

You have caught 1 fish(es)

Press Q to back to Games_Menu

Your Choice:

Car Racing Game

Once the users choose a track, the program will open the file and get information and store in the arrays. This program is almost the same with fishing game but it move the car when correct but not catch fish. Green car is user car while the yellow one is computer (correct answer).

1. Sample output on screen

A track and the cars

Lap Number (Question number)

Move (Question)

Choices

Answer space and check

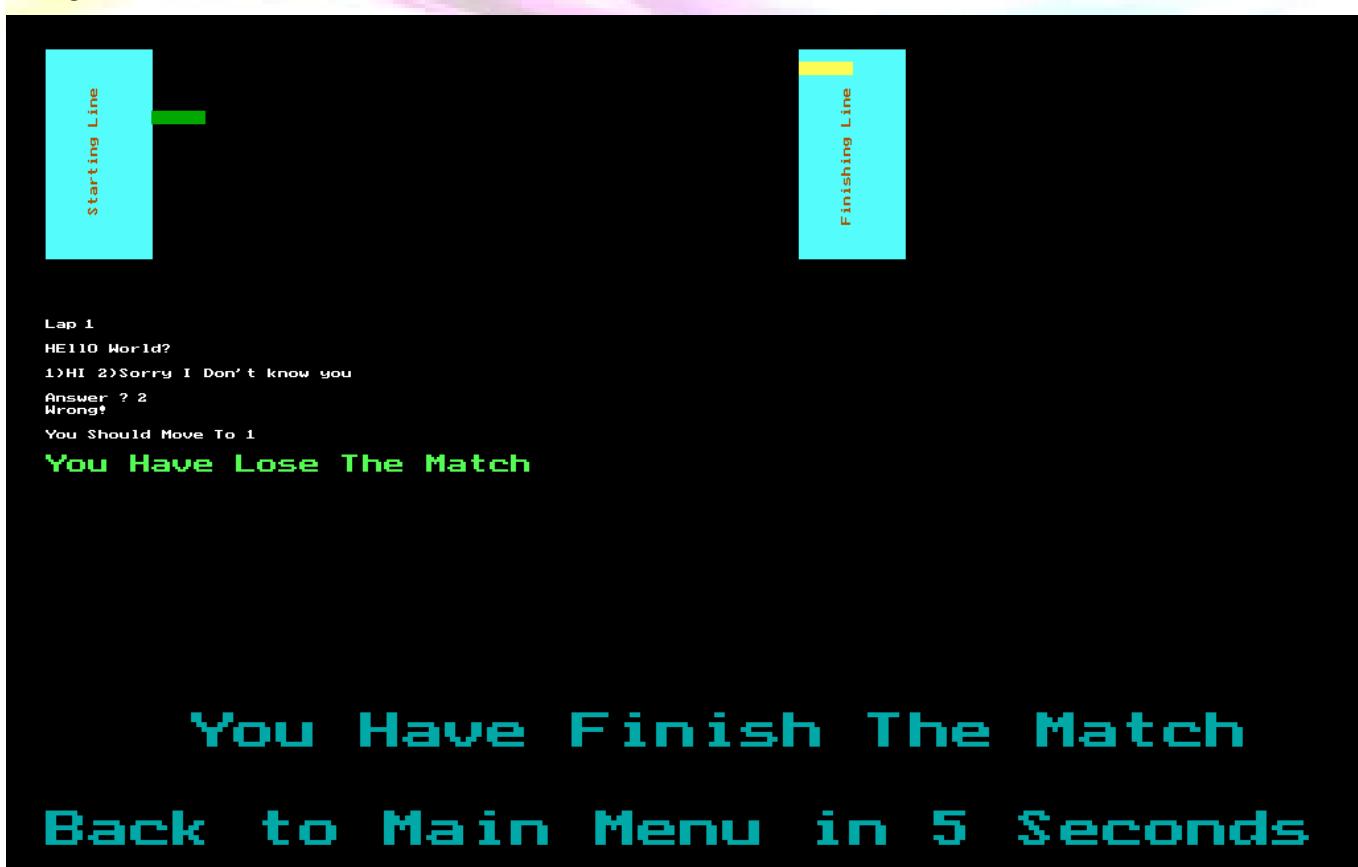
Win or not

Sample output on screen:

Correct answer and win:



Wrong Answer and lost:



The output above is the sub-program after choosing set, track etc. Therefore, the list of game resource is not shown. Here is the output sample before getting in the game

38

Sample output: (Vocabulary Guessing Game)

```
Topic Menu  
1:Vocab 1  
2:Vocab 2  
3:Vocab 3  
4:Game Menu  
Your Choice:
```

Sample output: (Matching Game)

```
Card Set Menu  
1:Set 1  
2:Set 2  
3:Set 3  
4:Game Menu  
Your Choice:
```

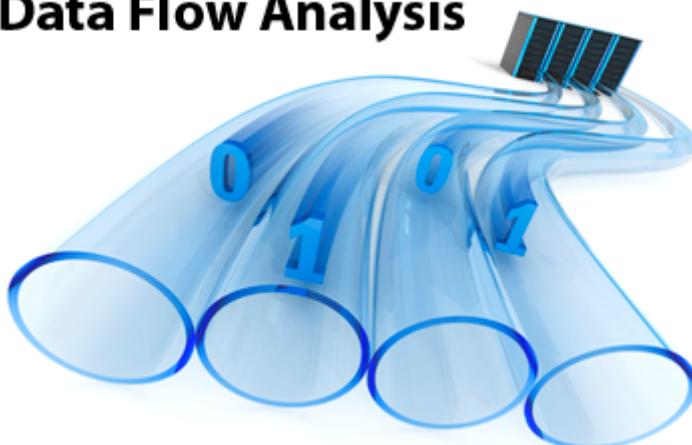
Sample output: (Fishing Game)

```
Location Menu  
1:Location 1  
2:Location 2  
3:Location 3  
4:Game Menu  
Your Choice:
```

Sample output: (Car Racing Game)

```
Track Menu  
1:Track 1  
2:Track 2  
3:Track 3  
4:Game Menu  
Your Choice:
```

Data Flow Analysis



*This program does not change the files much which mean does not need to output a lot files.

*The program mainly read, store and use the files.

Chapter 4 Implementation

4.1 Brief Description

- In this Chapter, I will discuss the implementation of the MC Analysis 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 and analysis report.



4.2 Data Structures

#After the Welcome Page pop up and user type to continue, the following panel will be used.

***The location (HDD) of the file of the program

The hard disk name (e.g. C:\) will be stored in global variable [HDD : string;]

The file of program is recommended to put in a hard disk that is connected to Internet/Intranet (e.g. P:\).

Login Panel

The program will check the account if it is value and stores the result for further use.

****The account will store as the name in the global variable, so the name in record MUST same as the account.*

{Global Variables}

```
valid_account: integer;
login_choice, c_name, Student_Class, Teacher_Class, status: string;
```

{Local Variables for Login procedures}

```
form, name, filename, pw, account, quit, ac : string;
login_file : text;
```

After the login_choice is typed by user, it uses to consider which procedure to be used:

Login_T or Login_S

As valid_account is 1 or 0 (Act as a flag here), it is an integer type.

For other variables, it allows almost unlimited length of words and need to be used later (some of them is use as filename);

form, name, pw are input by user first.

The filename is named by the form of the user.

```
filename:='F'+ form +' ICT.txt'; {Students}
```

```
filename:='F'+ form +'.txt'; {Teachers}
```

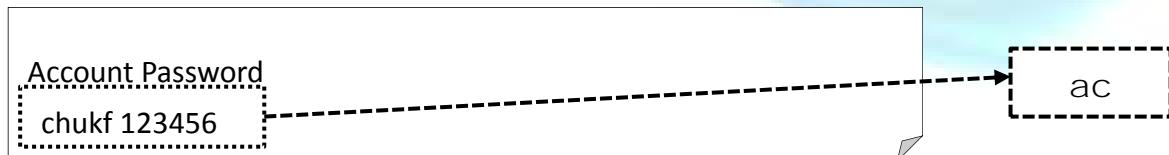
login_file is used to store the account and password file.

Account for searching:

```
account := name+ '' + pw;
```

The following example is a teacher:

F5.TXT



account is compare to ac until the file ends.

If the account is found valid_account returns to 1 (That is a value account)

And the following will be done

{Teachers}

```
Teacher_Class:='F'+ form;
```

```
status:='T';
```

{Students}

```
Student_Class:='F'+ form + ' ICT';
```

```
status:='S';
```

c_name is the name input by user (same as the account name)

After the account is found value, the program will restore the record of that class (discuss in next part). If the account is not value, the procedure is repeat until the user agree to quit (quit = Y).

Student Record Panel

I will use record type array here, which 1 array can store a lot of data.

{Global variables used}

type

recordtype = record

```
    Name: string;
    quiz1 , quiz2 , quiz3 :integer;
  end;
```

Student_Class, Teacher_Class, status : string;

Student:array[1..30] of recordtype;

s_num: integer;

Store Record

{Local variables}

record_file:text;

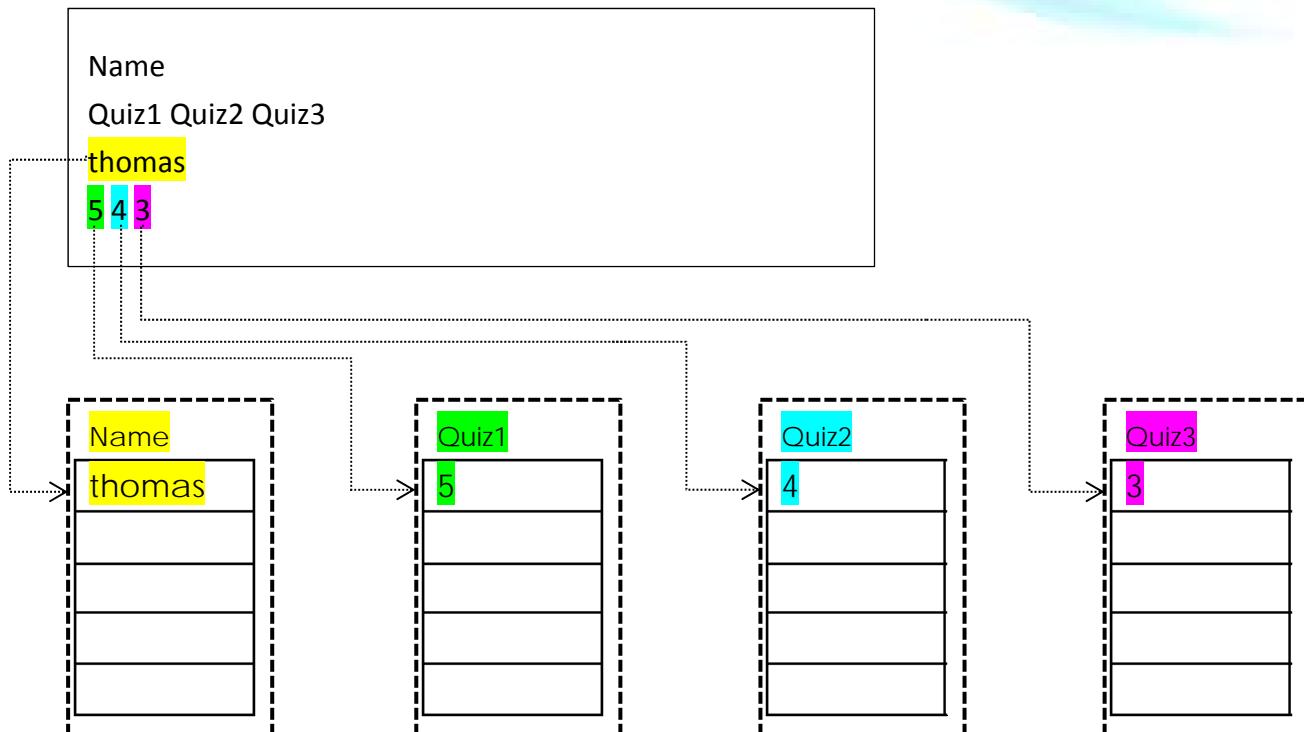
record_class:string;

student

Part for choosing the file:

```
if status<>'T'then
  record_class := Student_Class
else
  record_class := Teacher_Class + ' ICT';
```

e.g. Name[1], quiz1[1] ,quiz2[1]and quiz3[1] will store the corresponding data of the first F5 ICT.TXT



If the name in the record is same as the student name, the number is stored in s_num .

Show record

With considering the status, a specific record of a student or the record of whole class will be shown.

Students:

s_num is used in this part.

```
with Student[s_num] do
begin
  write(Name,' ');
  write(quiz1,' ');
  write(quiz2,' ');
  write(quiz3);
  writeln;
end
```

Teachers:

Show the records which are NOT EMPTY.

```
for i:= 1 to 30 do
with Student[i] do
  if Name <> " " then
    begin
      write(Name,' ');
      write(quiz1,' ');
      write(quiz2,' ');
      write(quiz3,' ');
      writeln;
    end;
```

Update record

It can be divided into 2 parts, update the local record (array) and update the record file.

Update record array

Once the user (student) did a quiz, if it is the first time for him to do (record score = -1) then the score will be stored in the array.

procedure Update_Record (quiz_done:char; score_record:integer);

```
with student[s_num] do
case quiz_done of
  '1': if quiz1 < 0 then quiz1 := score_record;
  '2': if quiz2 < 0 then quiz2 := score_record;
  '3': if quiz3 < 0 then quiz3 := score_record;
end;
```

Update record file

{Global variables}

Student_Class, Teacher_Class, status: string;

Student: array [1..30] of recordtype;

{Local variables}

U_Record_file: text;

U_Record_class: string;

```
if status<>'T' then
    U_Record_class := Student_Class
else
    U_Record_class := Teacher_Class + ' ICT';
```

The array of records will rewrite the OLD record file to update the record after the user quit ICT_Learner.

Resources Panel

This panel also has 3 sub-programs –read note, do exercise and do quiz.

Read Note

{Global variables}

choice: integer ;

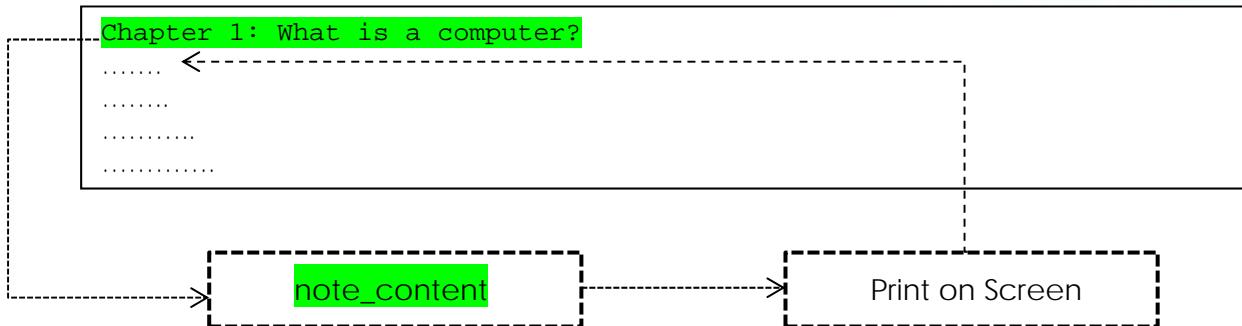
{Local variables}

note_content:string;

note_file:text;

After the user choose a note by a short procedure (discussed in previous chapter), the program will read the notes.

Note1.TXT



Do Exercises and Do Quizzes

This two parts are similar, I will use the quiz to be the example in this part.

{Local variables for exercises}

```
N,K,NC,L:integer;
QA:string;
ex_file:text;
QU,CH,AN :array[1..5] of string;
```

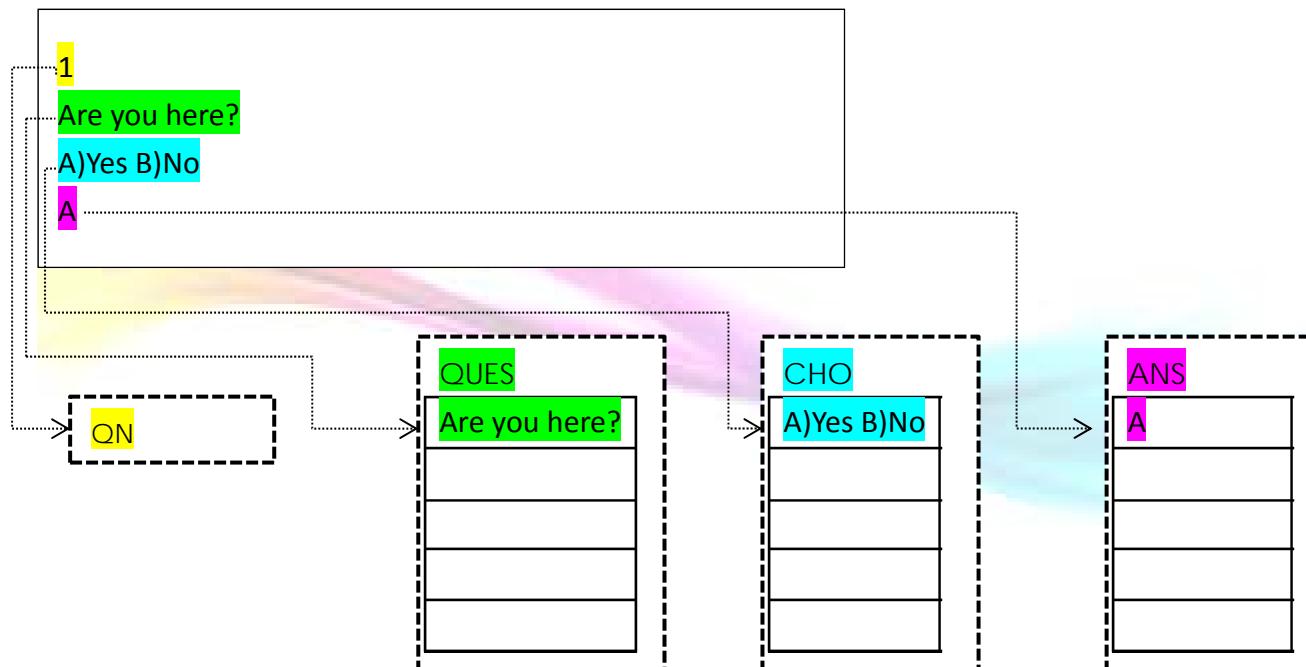
{Local variables for quiz}

```
quiz_file:text;
QN,QK,QNC,QL:integer;
QQA:string;
QUES,CHO,ANS :array[1..5] of string;
```

Here I used a parallel arrays , so that the number of question is clear.

Integers are used to stored number of questions, score, etc.

Quiz1.txt



```

readln(quiz_file,QN);
for QL := 1 to QN do
begin
  readln(quiz_file,QUES[QL]);
  readln(quiz_file,CHO[QL]);
  readln(quiz_file,ANS[QL])
end;
  
```

This procedure is similar to the fishing game and car racing game.

Comment Panel

This part read the content of comments and allows the users to add comments.

{Local Variables}

```
comment_file:text;
comments, comment_add, commentclass : string;
```

The string allow user to type any words (both number and characters).

```
if login_choice = 'S' then
    commentclass:=Student_Class
else
    commentclass:=Teacher_Class:
```

Take F5 ICT class as example:

F5 ICT.txt

admin:you can type your comment here

F5 ICT thomas : OK helloooooo<

Comments

Print on screen

It read comment until file ends and tell the user to give comment/ command.

After the user type a comment (Not the command of quit), the comment will be written in a new line of the comment file.

commentclass + c_name + comment_add

Back to Main Menu after user quit

Games Panel

There are 4 games in this panel.

The choice of which game and the sub topics to play will be chosen by the Choice Making procedure discussed in the first part.

Vocabulary Guessing Game

{Local variables}

Secret, Solution, topic_num : string;

Done : boolean;

Letter : char;

i : integer;

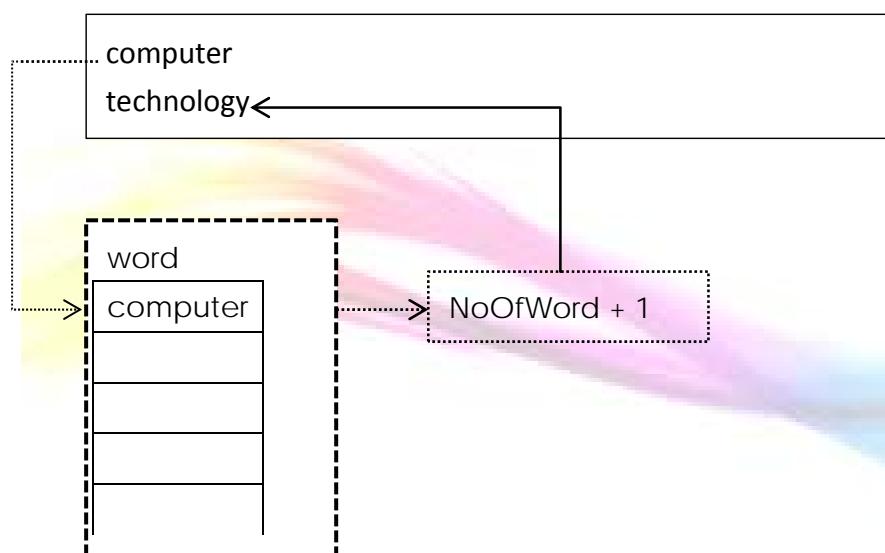
secretfile : text;

NoOfAttempts, NoOfWord : integer;

word : array[1..50] of string;

e.g. word[1] stored the first word.

Vocab1.TXT



Randomly choose a word from the word array to be a secret.

```
{---SelectSecret---}
assign(secretfile, HDD+'ICT_Learner\Games\Vocab_game\Vocab'+topic_num+'.txt');
reset(secretfile);
i := 0;
while not eof(secretfile) do
begin
  i := i + 1;
  readln( secretfile, word[i])
end;
close(secretfile);
NoOfWord := i;
randomize;
Secret := Word[random(NoOfWord)+1];
```

Matching Game

As this part will use graphic, global variables for storing the questions is used. And game is made up by different parts. (readIn has bugs in graphic mode for the developing tools)

The number of answer_match is equal to number of cards div 2. An array is used to store it.

The card content is stored in a 2- dimensional array, as there are 8 lines in one card and there are 9 cards.

{Global variables}

num_of_card, correct_match,num_of_pairs, fliped_times:integer;

card_content:array[1..9,1..8] of string;

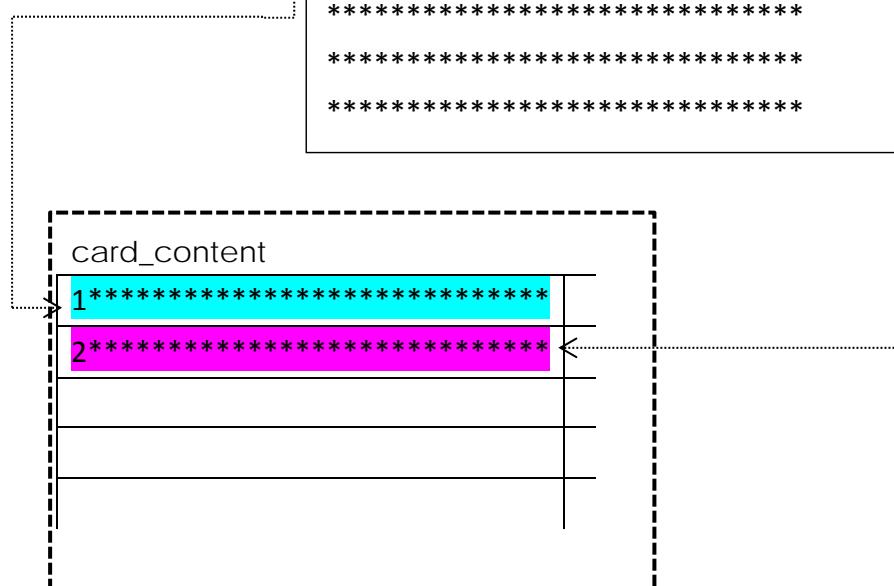
answer_match:array[1..4]of string;

e.g. answer_match[1] store the first available match

card_content:array[1,1] store the content of first line of first card

card_content:array[2,1] store the content of first line of second card

CardSet2.TXT



Fishing Game

{Local Variables}

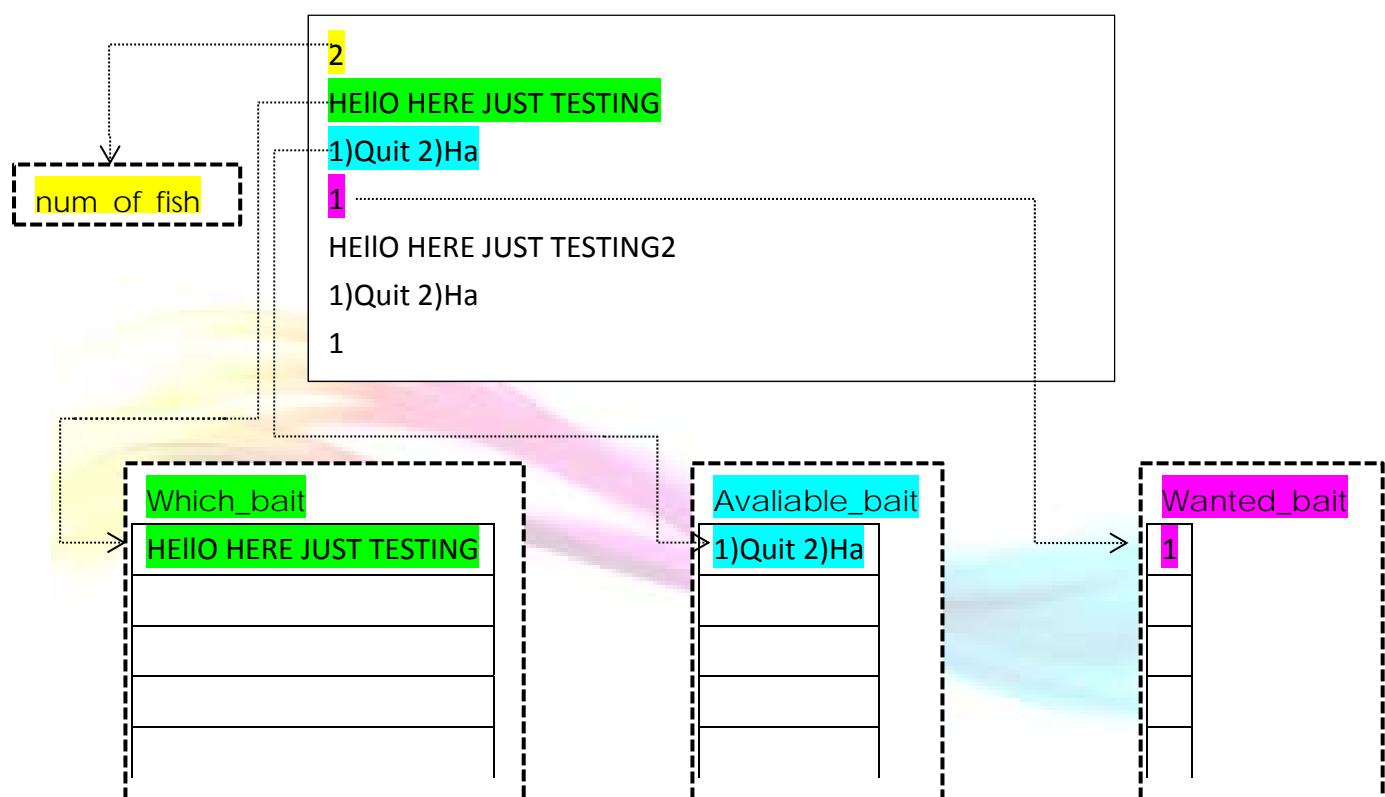
```
location_num:string;
Fish_file:text;
num_of_fish, FK, FC, FL, templocation_num :integer;
Selected_bait:string;
```

```
Which_bait,Available_bait,Wanted_bait :array[1..5] of string;
```

Arrays are used because there are more than 1 question.

After the user selects a location to play, it stores to local variables and find that file in the harddisk.

Location1.TXT



```
assign(Fish_file,'E:\ICT_Learner\Games\Fishing
Game\Location\Location'+ location_num+'.txt');
reset(Fish_file);
readln(Fish_file,num_of_fish);
for FL:= 1 to num_of_fish do
begin
  readln(Fish_file,Which_bait[FL]);
  readln(Fish_file,Available_bait[FL]);
  readln(Fish_file,Wanted_bait[FL])
end;
close(Fish_file);
```

After the questions are stored, a fish will randomly choose and print on screen.

49

Fish1.TXT

```
^m_ ````----..._<
` . . . . ` .
: . . . . ` .
; . . . . ` .
: `` : (@) ` .
` `` . =` . }` .
; . = ~ : . "
: `` . =` . ` .
: `` . . ` .
` . . . . ` .
! ``----"!` .
``
```



Car Racing Game

50

{Global Variable}

CN,CK,CNC,CL:integer;

CQ,CCH,CANS:array[1..5] of string;

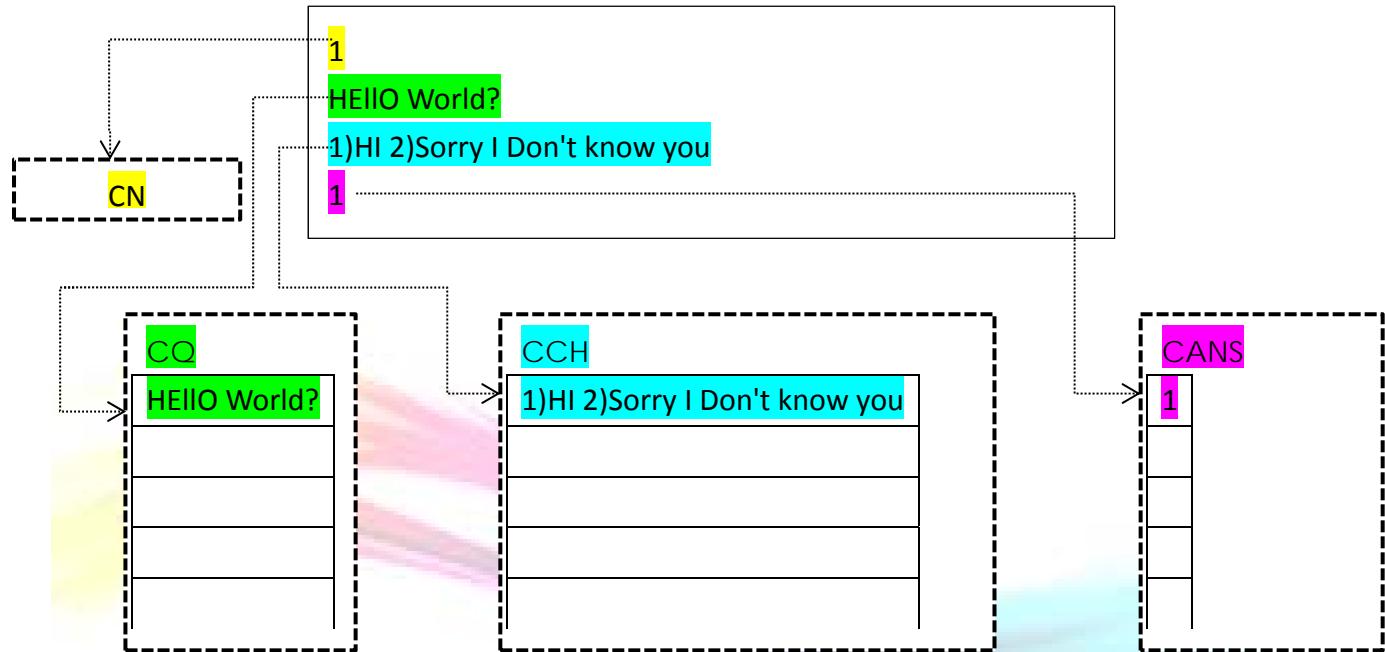
#The data structure of Fishing Game and Car Racing Game are similar as well as the quiz

{Local Variables}

track_file : text;

track_num : string;

Location1.TXT



```
reset(track_file);
readln(track_file,CN);
for CL := 1 to CN do
begin
  readln(track_file,CQ[CL]);
  readln(track_file,CCH[CL]);
  readln(track_file,CANS[CL])
end;
```

#The details of the procedure will be shown in the next sub-chapter 4.3

4.3 Procedures in the Program

I am going to describe some of the procedures which are not mention clearly in the previous parts as they don't have a data structure

Welcome page

```
{****-----Welcome Page-----****}
procedure Welcome_Page;
var graphkey:char;
begin
  GraphicsDriver := Detect;
  InitGraph(GraphicsDriver, GraphicsMode,"");
  SetColor(Green);
  settextstyle(TriplexFont,HorizDir,5);
  Outtextxy(70,300,'Welcome To ICT Learner');
  Delay(3000);
  SetColor(White);
  SetTextStyle(defaultfont, HorizDir,2);
  Outtextxy(270,400,'Type Any Key To Start Learning');
  graphkey:=readkey;
  if graphkey<>#0 then
    closegraph;
end;
```

This procedure prints big words on screen to welcome the user, the readkey is used to check if the user click a key. And it will continue.

Good Bye Page

```
begin
  GraphicsDriver := Detect;
  InitGraph(GraphicsDriver, GraphicsMode,"");
  SetColor(Green);
  settextstyle(TriplexFont,HorizDir,5);
  Outtextxy(230,300,'GOOD BYE');
  SetColor(White);
  SetTextStyle(defaultfont, HorizDir,2);
  Outtextxy(270,400,'Learn Next Time');
  delay(1000);
  closegraph;
end;
```

This is a pop up to say goodbye to the user.

Get Drive Name

```
{****---Get Drive Name---****}
procedure GetDriveName;
begin
  HDD:=(Copy((ParamStr(0)), 1, 2)+'\');
  writeln('The Directory Is Located In ', HDD);
end;
{****---End Of Getting Drive Name---****}
```

This procedure is quite short, however, it plays an important role in the program as it can increase the feasibility of the program. The user can put the main program folder in almost any harddisk they like.

#Almost all the choices are made by a choice making procedure and store choice as global variables.

Choice Making procedures

```
{****---Choice Making procedure---****}
procedure Make_choice;
begin
  write('Your Choice: ');
  readln(choice);
  writeln
end;
procedure Make_choice_chr;
begin
  write('Your Choice: ');
  readln(choice_chr);
  writeln
end;
```

choice: integer ; choice_chr :char;

Login Panel

The main part of the login procedure of both teachers and students:

```
repeat
  Write('What is your form?(e.g.4):');
  readln(form);
until (form = '4') or (form = '5')or (form = '6');
Write('What is your name?:');
readln(name);
c_name:= name;
Write('Password:');
readln(pw);
```

This procedure tells the user to login by typing different information.
All the information is stored as local variable to be used later.

Check the file if the account is valid:

```
while not eof(login_file) do
begin
  readln(login_file,ac);
  if account = ac then
  begin
    valid_account := 1;
    Student_Class:='F'+ form +' ICT';
    status:='S';
    Store_Record;
  end;
end;
```

(Student version)

Store_Record procedure will be shown in the next part

Store_record

```
while not eof(record_file) do
begin
i:=i+1;
with Student[i] do
begin
readln(record_file, Name);
readln(record_file, quiz1, quiz2, quiz3);
if Name = c_name then
s_num := i;
end;
end;
```

It stores the record to the array

Show record

```
for i:= 1 to 30 do
with Student[i] do
if Name <> " then
begin
write(Name,' ');
write(quiz1,' ');
write(quiz2,' ');
write(quiz3,' ');
writeln;
end;
```

(Teacher version)

All the records of students of that class are shown.

While for student with Student[s_num] do is used instead of with Student[i] do.

Update record

```

procedure Update_Record(quiz_done:char; score_record:integer);
begin
with student[s_num] do
case quiz_done of
'1': if quiz1 < 0 then quiz1 := score_record;
'2': if quiz2 < 0 then quiz2 := score_record;
'3': if quiz3 < 0 then quiz3 := score_record;
end;
end;
```

The parameter of this procedure will be input by the procedure of quiz.

Update record file

```

for i:= 1 to 30 do
begin
with Student[i] do
if Name <> " then
begin
writeln(U_record_file,Name);
writeln(U_record_file, quiz1 , ' ', quiz2 , ' ', quiz3 , ' ');
end;
end;
```

Rewrite the whole file with the restored array.

Resources Panel

A command will be made to back to note menu by user.

This part is similar to all quiz and exercise parts, therefore it will not be discussed again in next parts.

```

writeln('Press Q to back to Notes List');
Make_choice_chr;
if choice_chr = 'Q' then
begin
Clrscr;
if choiceC = 'Complusory' then
    Complusory_Notes_List
else Elective_Notes_List;
end
```

Read Note

```
reset(note_file);
while not eof(note_file) do
begin
  readln(note_file,note_content);
  writeln(note_content);
end;
close(note_file);
```

56

Read one line and write one line until the end of file.

Do Exercises and Do Quizzes

The procedure of storing the question, answer, choice is discussed in the previous chapter.

Here is the procedure of doing exercise/quiz.

```
for K:= 1 to N do
begin
  writeln('Question ',K);
  writeln(QU[K]);
  writeln(CH[K]);
  repeat
    write('Answer ? ');
    readln(QA);
  until QA<>"";
```

This part of procedure is for doing exercise or quiz.

The answer will be shown in the exercise but not in quiz.

Moreover, the score of quiz will be stored as mentioned before.

```
writeln("You have ', QNC, ' mark(s)");
Update_Record(quiznum[1], QNC);
writeln('Answer will be given by teachers');
```

Comment Panel

The procedure is showing how to read and write a comment

Read

```
Assign(comment_file,HDD+'\ICT_Learner\Comment\' +commentclass+'.TXT');
reset(comment_file);
while not eof(comment_file) do
begin
readln(comment_file,comments);
writeln(comments)
end;
Close(comment_file);
```

Write

```
Assign(comment_file,HDD+'\ICT_Learner\Comment\' +commentclass+'.TXT');
Append(comment_file);
repeat
writeln('What comment you want to add?(Type <Q> to Exit)');
readln(comment_add);
if comment_add <> 'Q' then
begin
Writeln(comment_file,commentclass+' '+c_name+': '+ comment_add);
end
...
Close(comment_file);
```

"Append" command is used as it won't delete all the previous comment like rewrite.

And append comment can add things in the last line of the file which is equal to the last comment.

Games Panel

58

****As 2 of the games is in graphic, get into the graphic mode is needed(Both Matching game and Car racing game)

{Graphic}{Global Variables}

GraphicsDriver, GraphicsMode: smallint

```
Writeln('Entering The Game, please wait... ');
GraphicsDriver := Detect;
InitGraph(GraphicsDriver, GraphicsMode, "");
```

#The ways to get the secret is mention in the previous chapter, I will focus on the procedure playing games this part

Vocabulary Guessing Game

```
{---LetterGuess---}
repeat
begin
  write('Please input the letter you guess: ');
  readln(letter);
  for i := 1 to length(Secret) do
    if Letter = Secret[i] then
      Solution[i] := Letter;
    writeln(Solution);
  NoOfAttempts := NoOfAttempts - 1;
  writeln('No. of Attempts remaining = ', NoOfAttempts);
{---WinCheck---}
  Done := (Solution = Secret) or (NoOfAttempts = 0);
end
until Done = true;
if Solution = Secret then
  writeln('You win!!')
else
  writeln('You lose!!');
writeln;
```

This procedure shows a platform to be used to play the game.

Matching Game

59

```
setfillstyle(emptyfill,0);
Bar(30,530,600,630);
moveto(30,530);
setcolor(white);
SetTextStyle(DefaultFont, HorizDir,1);
outtext('The 1st card you want to flip:');
cardchoice1st:= readkey;
outtextxy(275,530,cardchoice1st);
moveto(30,550);
Clear_card(cardchoice1st);
Write_Card_Content(Ord(cardchoice1st)-48);
outtext('The 2nd card you want to flip:');
cardchoice2nd:= readkey;
outtextxy(275,550,cardchoice2nd);
Clear_card(cardchoice2nd);
write_Card_Content(Ord(cardchoice2nd)-48);
user_match:=cardchoice1st+'-'+cardchoice2nd;
user_match2:=cardchoice2nd+'-'+ cardchoice1st;
i:=0;
```

Bar uses to clear the screen.

Readkey is used to read the key that user press which is the card to be flip

Write_Card_Content(card_num:integer); is used to write the content on the screen after the player flip the card.(Read the full program for reference).

```
setfillstyle(solidfill,yellow);
setcolor(yellow);
case fill_card of
1:Bar(30,30,270,180);
2:Bar(290,30,530,180);
3:Bar(550,30,790,180);
4:Bar(30,200,270,350);
5:Bar(290,200,530,350);
6:Bar(550,200,790,350);
7:Bar(30,360,270,510);
8:Bar(290,360,530,510);
9:Bar(550,360,790,510);
end;
```

This part use to draw the card and allow the users to choose.

Clear card is drawing the same shape of bar but different color (background color)

Fishing Game

60

Fishing game is similar to exercise and quiz but a fish will be shown every question for the user to catch.

After the fish file is chose, the fish will be draw on the screen

```
Randomize;  
TextColor(random(14)+2);  
while not eof(Fish_file) do  
begin  
write('');  
readln(Fish_file,Fish_line);  
writeln(Fish_line);  
end;
```

The variables are described in the previous chapter.

```
for FK:= 1 to num_of_fish do  
begin  
Draw_Fish;  
TextColor(White);  
writeln('Fish ',FK);  
writeln(Which_bait[FK]);  
writeln(Available_bait[FK]);  
repeat  
write('Answer ? ');  
readln(Selected_bait);  
until Selected_bait<>"";  
if Selected_bait = Wanted_bait[FK] then  
begin  
writeln('You Caught The Fish!');  
writeln;  
FC := FC + 1  
end  
else  
begin  
writeln('You Missed The Fish!');  
writeln('The Fish Wanted Bait ',Wanted_bait[FK]);  
writeln;  
end;
```

This part is the part of procedure for fishing

Car Racing Game

This game need to use graphic mode, so, the game will start with a procedure of getting into the graphic mode as stated before.

```
procedure Draw_Car_u(A:integer;B:integer);
Begin
  setfillstyle(solidfill,Green);
  setcolor(Green);
  Bar(A,B,A+40,B+10);
End;
procedure Draw_Car_c(A:integer;B:integer);
Begin
  setfillstyle(solidfill,yellow);
  setcolor(yellow);
  Bar(A,B,A+40,B+10);
End;
```

These 2 procedures are used to move the cars.

```
if User_Move = CANS[CK] then
begin
  outtextxy(30,320,'Right!');
  CNC := CNC + 1;
  ucpx:=ucpx+(490 div CN);
  ccpx:=ccpx+(490 div CN);
end
else
begin
  outtextxy(30,320,'Wrong!');
  outtextxy(30,340,'You Should Move To '+CANS[CK]);
  ccpx:=ccpx+(490 div CN)
end;
```

After the user answers the question, which procedure is like the previous in fishing game, these procedure checks if the answer is correct and decide how many step to move by the user.

Finally, for the Main Program, as the program used a lot of procedures the main program is short.

62

```
{-----Main Program-----}  
Begin  
    Welcome_Page;  
    writeln('LETS START LEARNING ICT');  
    writeln('ENJOY');  
    writeln('Loading...');  
    Delay(1000);  
    GetDriveName;  
    Delay(500);  
    Login;  
    Update_Record_File;  
    Good_Bye_Page;  
End.
```

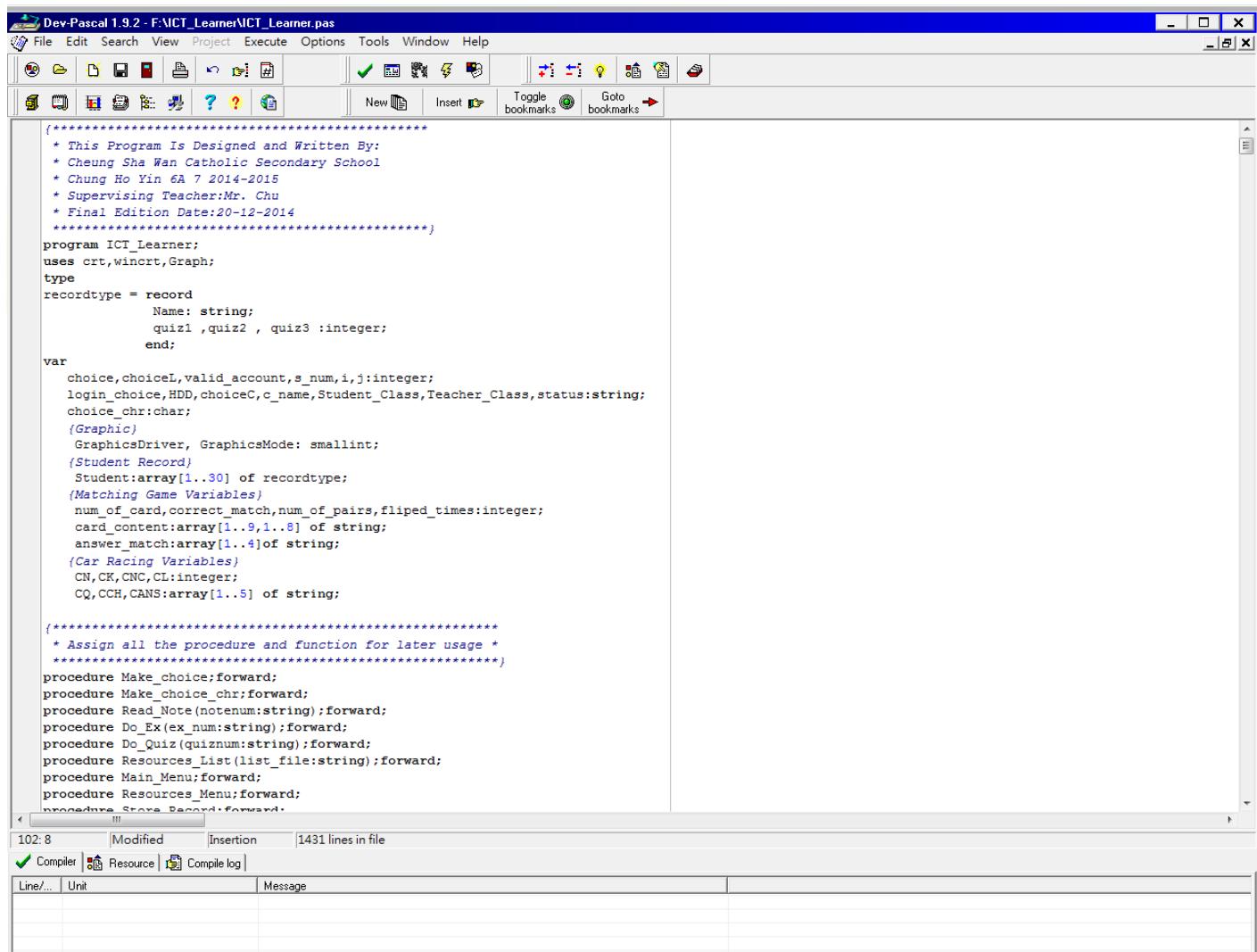
This chapter has been showing the most important, useful and major procedure. For the minor procedures, please read the complete program code (Appendix 1)



4.4 Program Coding

- Briefly describe the process of program coding (writing).
 - By using Pascal programming language and tool Dev-Pascal.
 - Writing the source program and then compile it into the object program (ICT_Learner.exe)
 - Filename of the source program is ICT_Learner.PAS
- Filename of the object program is ICT_Learner.EXE
- Folder name of the program is ICT_Learner, the folder ICT_Learner need to be place in the root of the hard disk. (e.g.P:)
- A Help guide will be provided in the Appendix Section.
- The complete program code is placed in the Appendix Section. Refer to Appendix 1 for the complete program code.

Screenshot of the developing tool:



The screenshot shows the Dev-Pascal 1.9.2 IDE interface. The title bar reads "Dev-Pascal 1.9.2 - F:\ICT_Learner\ICT_Learner.pas". The menu bar includes File, Edit, Search, View, Project, Execute, Options, Tools, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and Execute. Below the toolbar is a row of small icons for different functions. The main window displays the Pascal source code for the program ICT_Learner. The code includes comments at the top about the program's purpose and authorship. It defines record types for student records, matching game variables, and car racing variables. Procedures for making choices, reading notes, doing exercises, and managing resources are also defined. The code is well-organized with multiple sections separated by asterisks. At the bottom of the IDE, there is a status bar showing "102: 8 Modified Insertion 1431 lines in file" and tabs for Compiler, Resource, and Compile log.

```

***** This Program Is Designed and Written By:
***** Cheung Sha Wan Catholic Secondary School
***** Chung Ho Yin 6A 7 2014-2015
***** Supervising Teacher:Mr. Chu
***** Final Edition Date:20-12-2014
***** */

program ICT_Learner;
uses crt,winCRT,Graph;
type
recordtype = record
  Name: string;
  quiz1 ,quiz2 , quiz3 :integer;
end;
var
  choice,choiceL,valid_account,s_num,i,j:integer;
  login_choice,HDD,choiceC,c_name,Student_Class,Teacher_Class,status:string;
  choice_chr:char;
  {Graphic}
  GraphicsDriver, GraphicsMode: smallint;
  {Student Record}
  Student:array[1..30] of recordtype;
  {Matching Game Variables}
  num_of_card,correct_match,num_of_pairs,fliped_times:integer;
  card_content:array[1..9,1..8] of string;
  answer_match:array[1..4]of string;
  {Car Racing Variables}
  CN,CK,CNC,CL:integer;
  CQ,CCH,CANS:array[1..5] of string;

/*
 * Assign all the procedure and function for later usage *
 *****/
procedure Make_choice;forward;
procedure Make_choice_chr;forward;
procedure Read_Note(notenum:string);forward;
procedure Do_Ex(ex_num:string);forward;
procedure Do_Quiz(quiznum:string);forward;
procedure Resources_List(list_file:string);forward;
procedure Main_Menu;forward;
procedure Resources_Menu;forward;
procedure Score_Record;forward;

```

4.5 Program Execution

1. Program file: ICT_Learner.EXE
2. Data file to be prepare for input:
 - o The text files storing the account files, lists of resources, notes, exercises, quizzes and the game resources:
 - o Login Panel
 - F4.txt,F5.txt and F6.txt (Teacher)
 - ◆ Stored in "ICT_Learner\Login\Teacher_Accounts"
 - F4 ICT.txt,F5 ICT.txt and F6 ICT.txt(Student)
 - ◆ Stored in "ICT_Learner\Login\Student_Accounts"
 - o Resources Panel
 - Note_C_List.txt, Ex_C_List.txt and Quiz_C_List.txt (C refer to compulsory , E for elective)
 - ◆ Lists of resource stored in "ICT_Learner\Resources"
 - Note1.txt, Notes2.txt etc.
 - ◆ Note txt stored in "ICT_Learner\Resources\Notes\Complusory"
 - Ex1.txt, Ex2.txt etc.
 - ◆ Exercise txt stored in "ICT_Learner\Resources\Exercises\Complusory"
 - Quiz1.txt,Quiz2 ,Quiz3
 - ◆ Quizzes txt stored in "ICT_Learner\Resources\Quizzes\Complusory"

Only compulsory resources are available now.

- o Comment Panel
 - F4 ICT.txt,F5 ICT.txt and F6 ICT.txt
 - ◆ Stored in "ICT_Learner\Comment"
- o Games Panel
 - Vocab Guessing Game
 - Vocab1, Vocab2 etc.
 - Stored in "ICT_Learner\Games\Vocab_game"
 - Matching Game
 - CardSet1.txt, CardSet2.txt, CardSet3.txt,
 - Stored in "ICT_Learner\Games\Matching Game"
 - Fishing Game
 - Fish1.txt, Fish2.txt, etc.
 - Stored in "ICT_Learner\Games\Fishing Game\Fishes"
 - Location1.txt, Location2.txt, Location3.txt
 - Stored in "ICT_Learner\Games\Fishing Game\Location"
 - Car Racing Game
 - Trackset1.txt, Trackset2.txt, Trackset3.txt,
 - Stored in "ICT_Learner\Games\Car Racing Game"

Screenshots of the user interface:

65

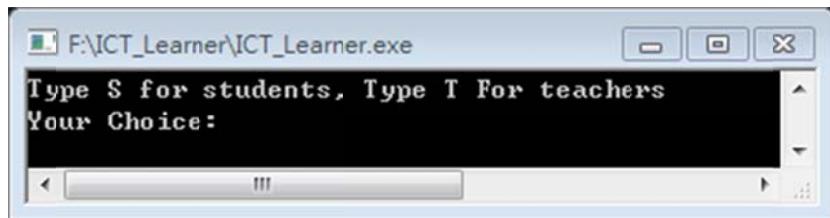
Welcome page:



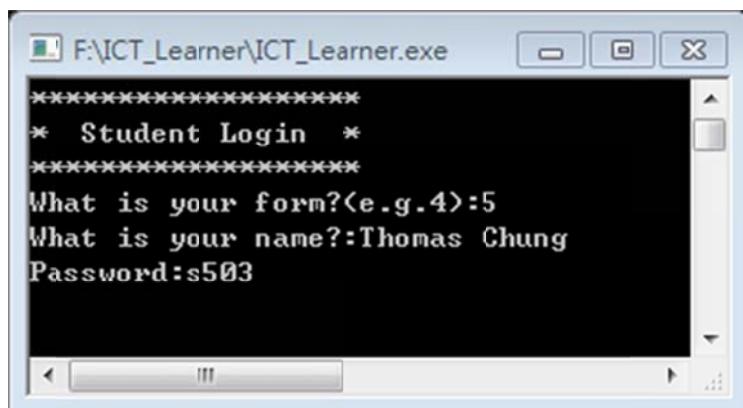
This is a page that will be pop up directly after the user starts the program, the words are printed on screen in green color in order to attract the user. And wake the user up. Also, the interface tells to user to type a key, it makes sure that the user is concentrating on the screen.

After this page, a login platform will be shown.

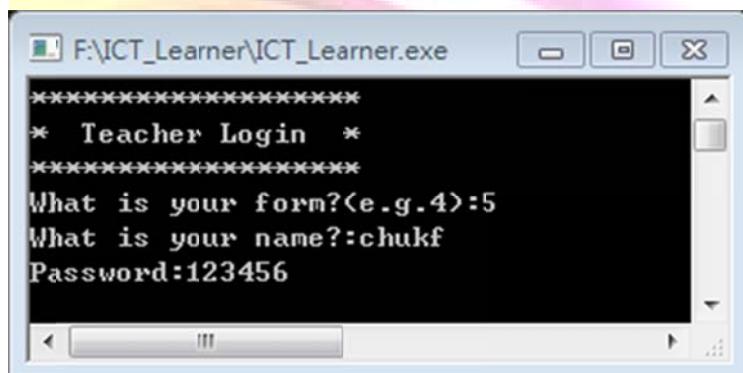
Login Platform:



This simple statement can help the computer to classify the user between students and teachers, to give the relevant panels and resources for the users.



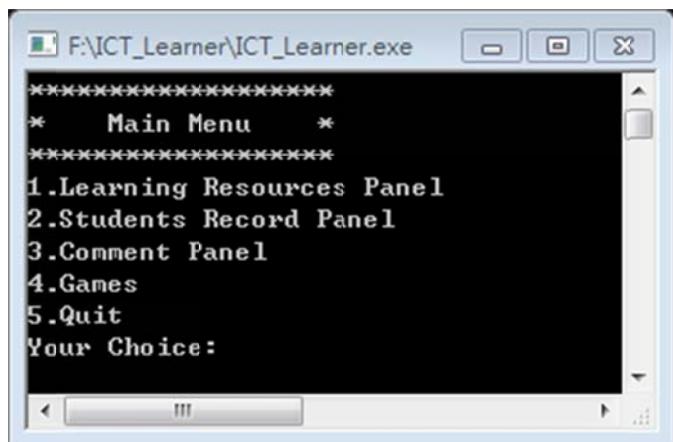
This is the login process for the students, the user is required to type in form, name and password. After the user login, there will be different functions.



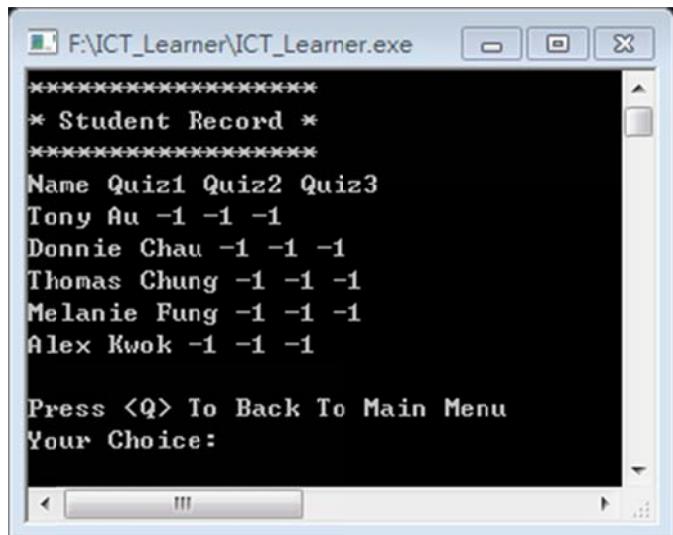
This is the login process for the teachers, the user is required to type in form, name and password. After the user login, there will be different function.

After the users login, a main menu will be shown.

Main Menu



Record Panel:

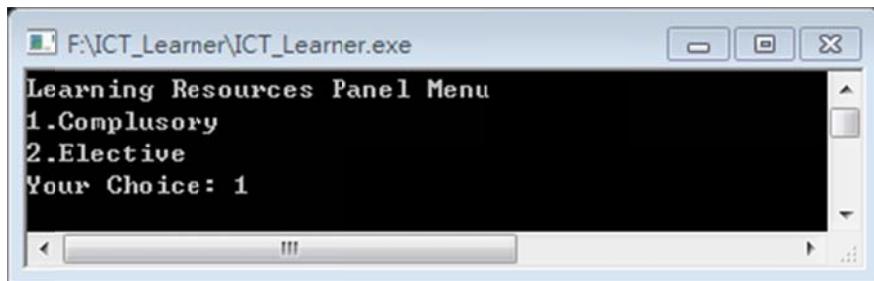


This panel shows the records of the student for the student version, while the records of whole class will be shown for teachers.

The record is stored in the login process. The temporary record will be stored in the variable for further updates. The records will be updated to the files when the users log in out.

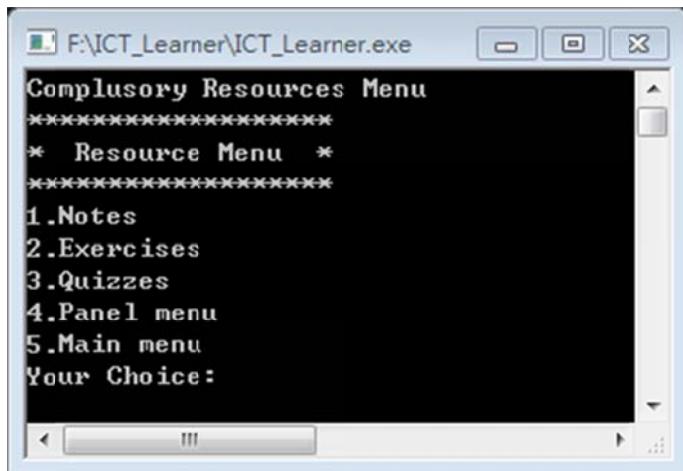
Resources Panel:

The panel of students and teachers is a bit different.

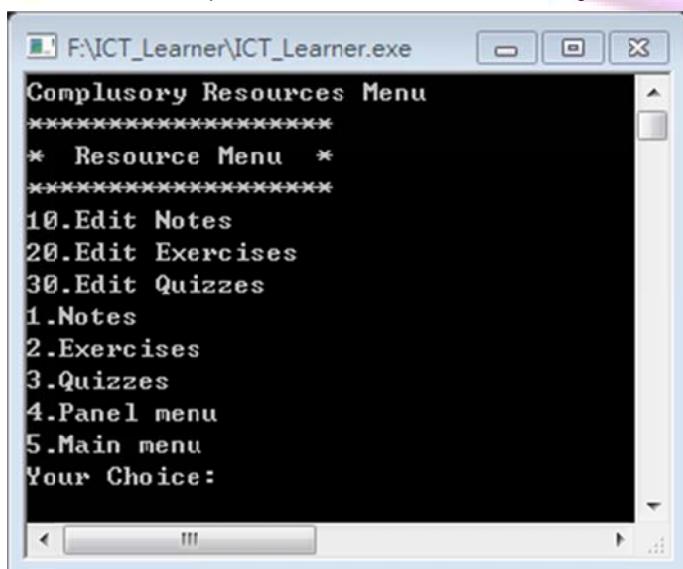


With considering the further development of the program, it is allowed to choose from compulsory and elective part.

Compulsory Resources:



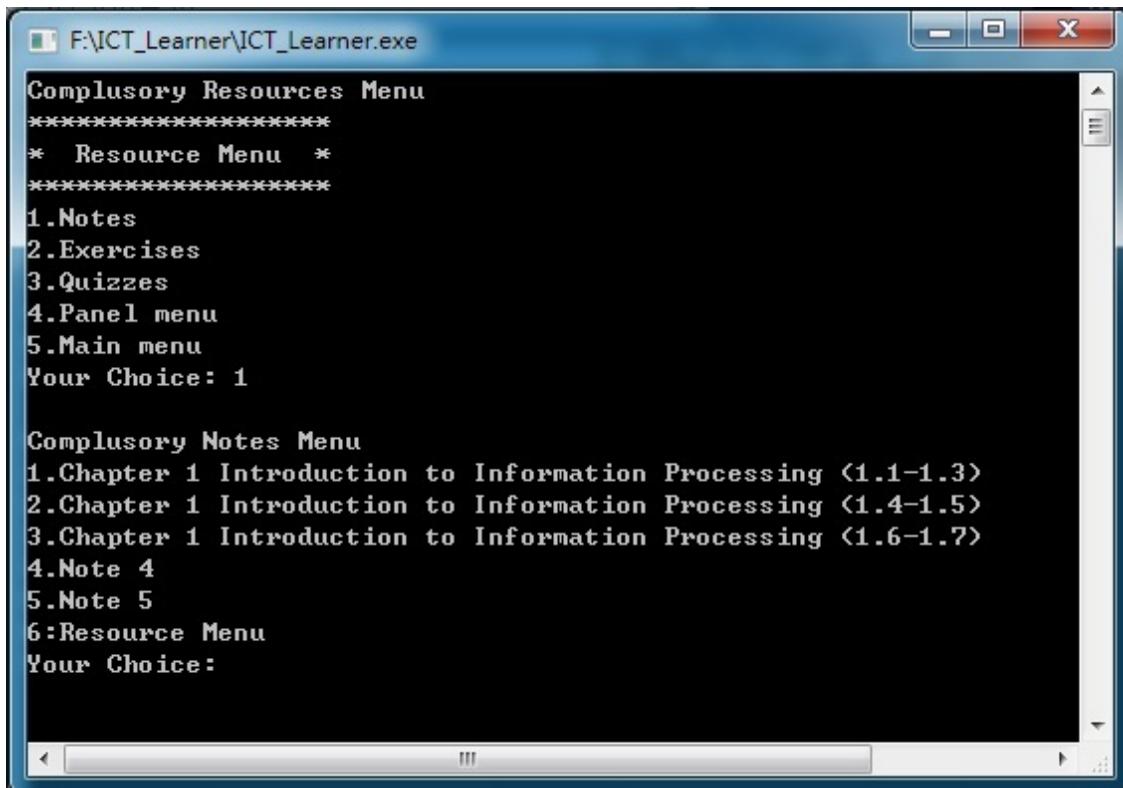
This is the panel for the students, they can choose from different resources to help learning ICT.



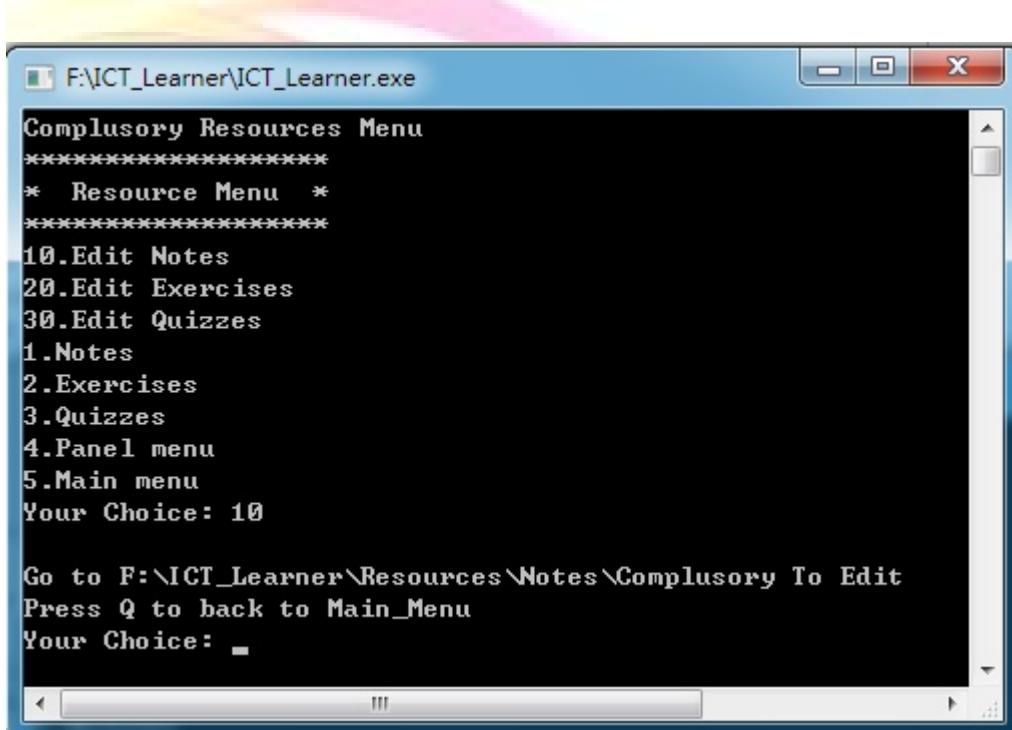
This is the panels for the teachers, the teachers have permission to edit the resources. The path of the resources will be given to the teachers if they choose to edit. They can edit the file directly by notepad.

Menu for different resources:

Notes menu:

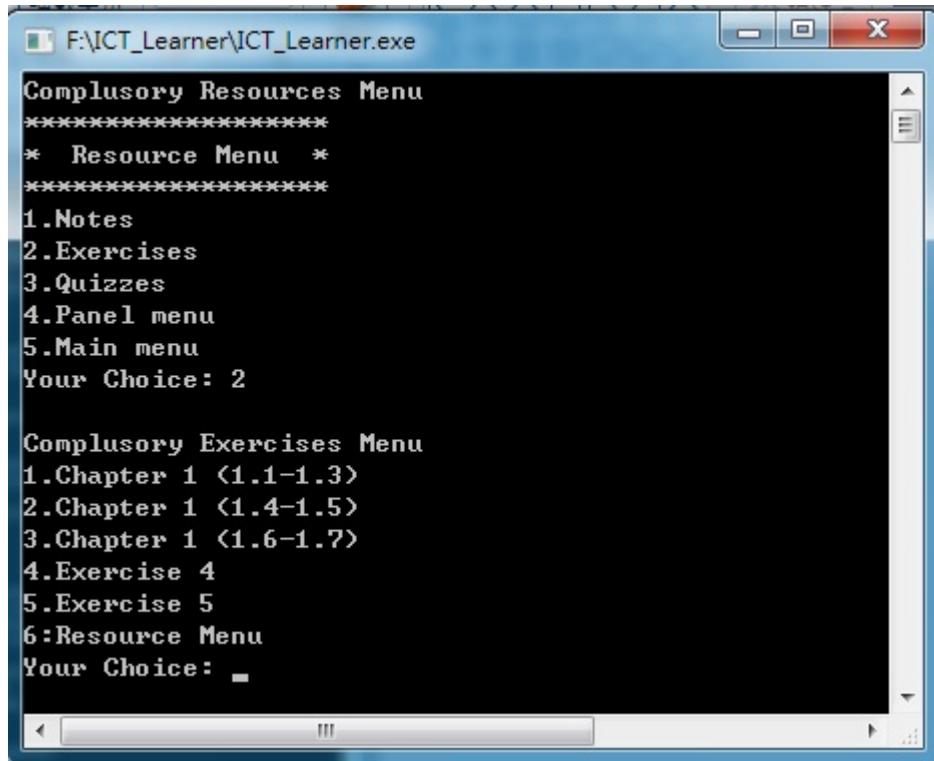


It is a page which shows the notes that the users can choose to read. The users can choose notes from different topics.



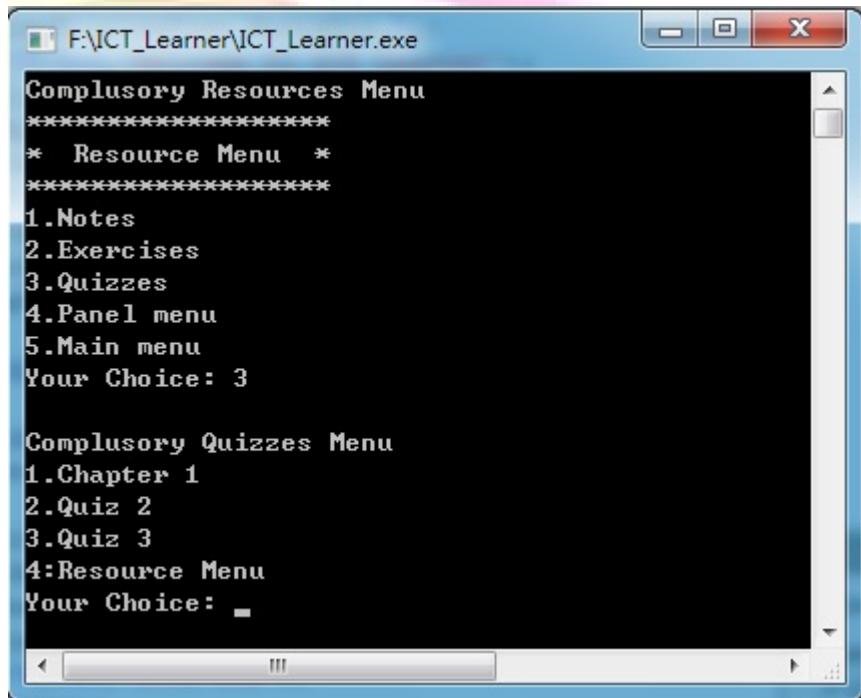
A page that shows the location of the files of that they want to edit.

Exercises menu:



It is a page which shows the exercises that the users can choose to read. The users can choose exercises from different topics.

Quiz menu:



It is a page which shows the quizzes that the users can choose to read. The users can choose quizzes from different topics.

Different page of using resources:

Reading notes:

The screenshot shows a window titled 'F:\ICT_Learner\ICT_Learner.exe'. The content is a note file with the following sections and text:

- * Chapter 1 Introduction to Information Processing ***
- 1.1 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 complicated calculations
 - Produce the calculated results on different media
 - Input-Process-Output cycle usually involves operations and hardware.
 - Input
 - Enter data and commands into a computer via one or more input devices.
 - Process
 - The CPU (Central Processing Unit) works on the data according to the commands provided in the input stage.
 - Output
 - The computer shows the processed result on the output devices specified in the input stage.
- 1.2 Use of Stored Programs**
 - Computers will not function if no command is given.
 - A program is a sequence of commands which instruct the computer to complete a specific task. The computer will fetch and execute the commands one by one.
 - Stored programs enable computers to do tasks automatically, repeatedly and accurately.
 - Stored programs saved in the read-only memory (ROM) of many home appliances and machines with small circuit board
 - Stored programs are used in many home appliance and machines in our daily life.
- 1.3 Difference between Data and Information**
 - Data:**
 - A collection of raw facts that are not organized.
 - Has no meaning on their own.
 - Information:**
 - Data that has been organized
 - Meaningful and useful for decision making

At the bottom, there is a message: 'Press Q to back to Notes List' and 'Your Choice: _'.

This is a page to read a note file and show it on the screen.

In this page, the students can read the notes and learn ICT and improve their performance.

The users can get back to notes menu by pressing <Q>.

Doing exercises:

The screenshot shows a window titled 'F:\ICT_Learner\ICT_Learner.exe'. The content is an exercise with the following questions:

- There are 5 question(s) in this exercise
- Question 1**
Data must be processed and organized to become information?
A>True B>False
Answer ? A
Right!
- Question 2**
Enter data and commands into a computer, what is it called?
A>Input B>Process C>Output
Answer ? A
Right!
- Question 3**
Stored programs must be stored in optical or magnetic storage devices?
A>True B>False
Answer ? B
Right!
- Question 4**
Give the full name of ROM.
Type the answer!
Answer ? read-only memory
Right!
- Question 5**
Stored programs enable computers to do tasks _____, repeatedly and accurately.
Fill in the blank by only ONE word
Answer ? automatically
Right!

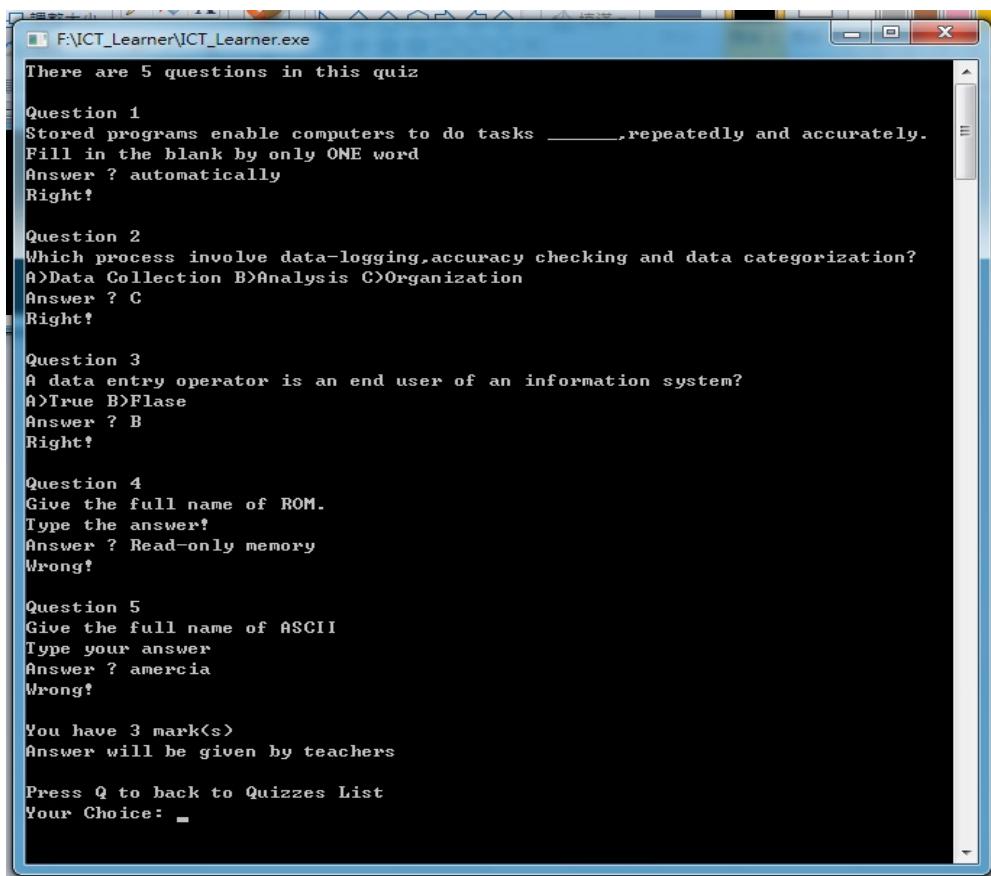
At the bottom, there is a message: 'You have 5 mark(s)' and 'Press Q to back to Exercises List' and 'Your Choice: _'.

This is a page to read an exercise file and show it on the screen.

In this page, the students can test their understanding towards the topics by completing the exercise

The users can get back to exercises menu by pressing <Q>.

Doing quizzes:



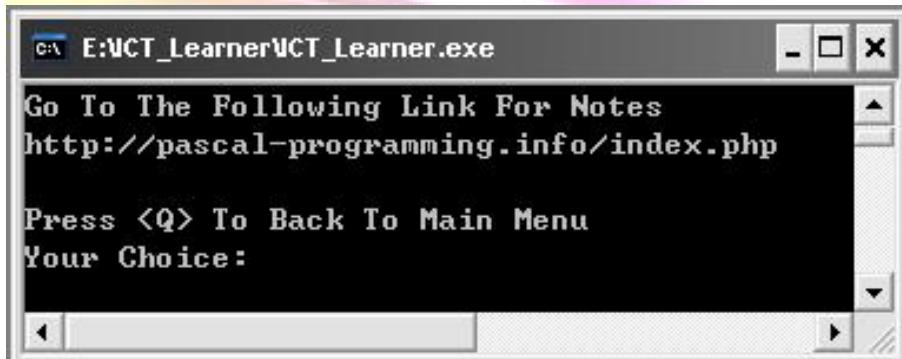
This is a page to read a quiz file and show it on the screen,

In this page, the students can do a quiz while the answer will not be shown in the program. Their marks will be updated to a mark file.

The teachers can check the learning progress and the potential of students.

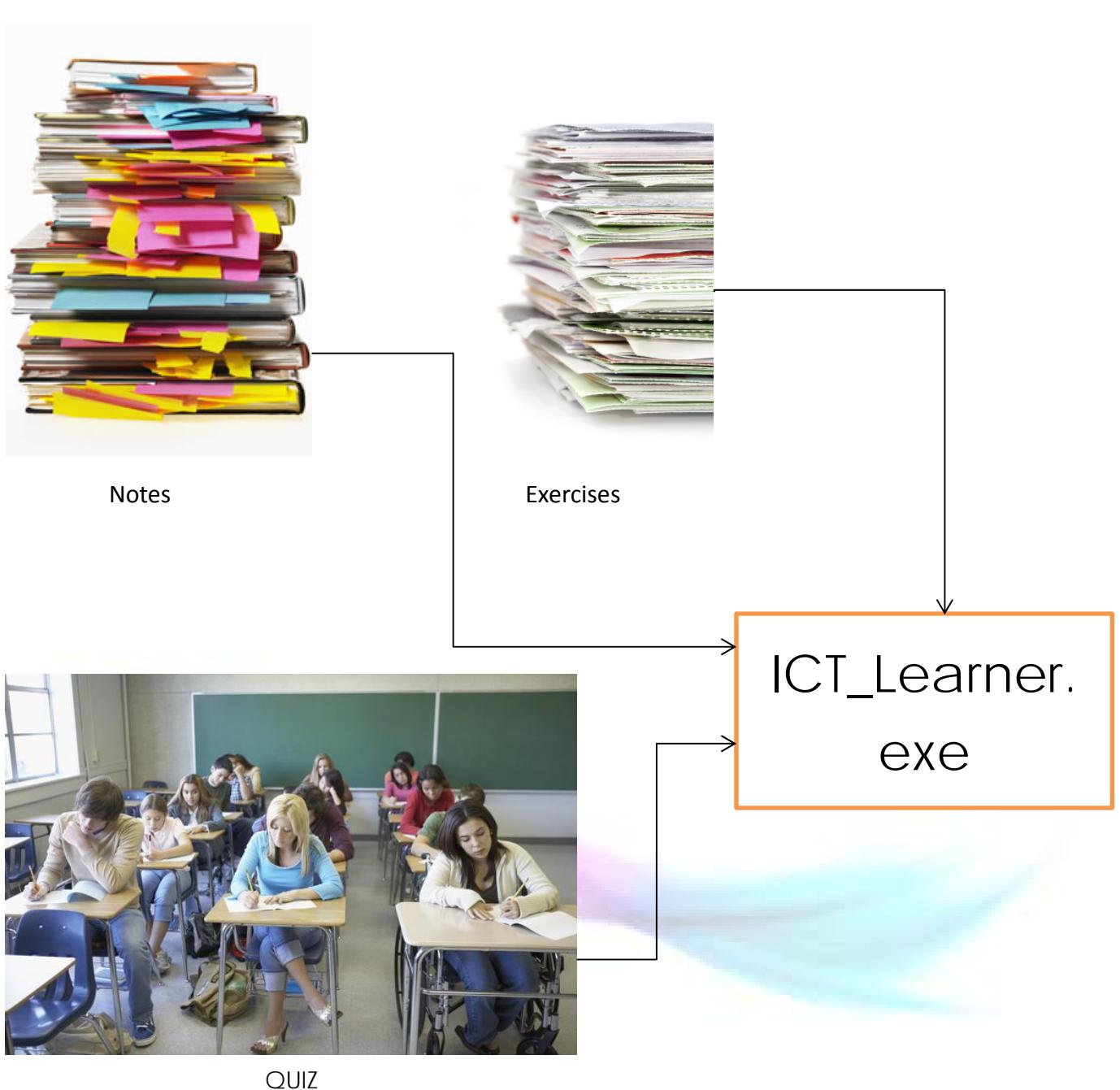
The users can get back to quizzes menu by pressing <Q>.

Elective Part:

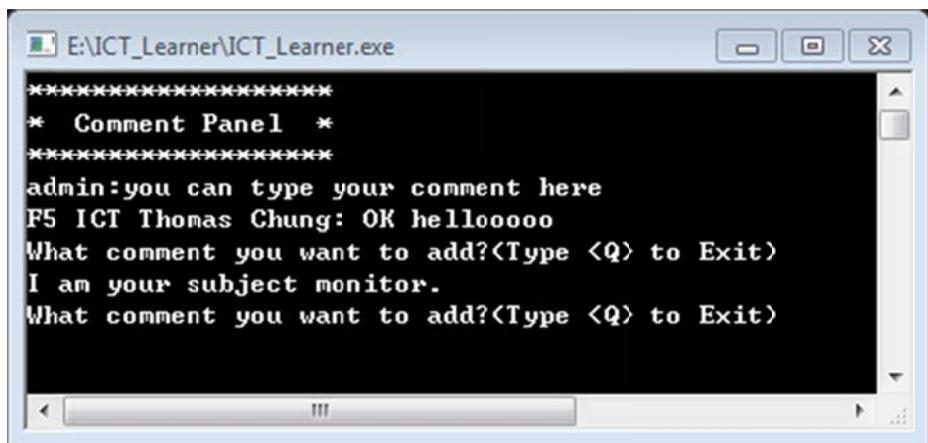


This is just a temporary page for the elective part. A websites contain the resource of 2D software development of Pascal is recommend to the students.

The users can get back to main menu by pressing <Q>.

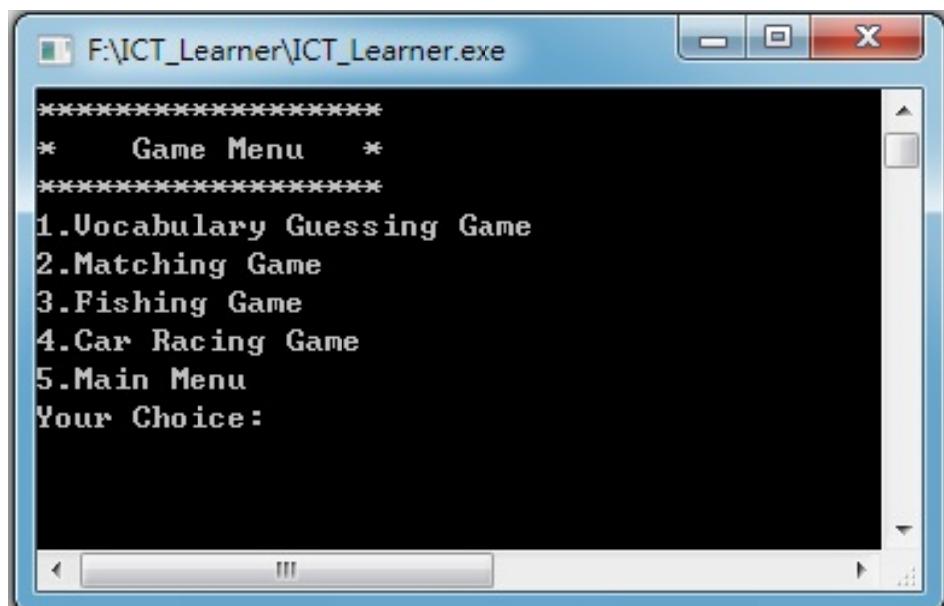


Comment Panel:



This panel is designed for the students and teachers to do academic exchange and help each other. The comment will be updated after the user type in a comment. The teachers are responsible for cleaning the past message after backing up them.

Game Menu:

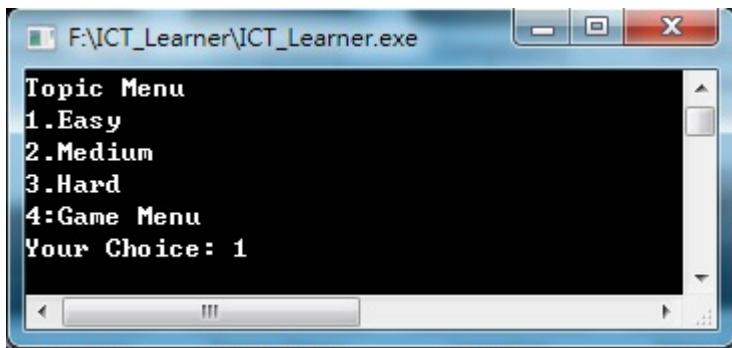


This is the part that provides different game for the users to learn from pleasure. They will be able to choose which level to play after they choose a game.

Some of the example of the game will be shown in the next page.

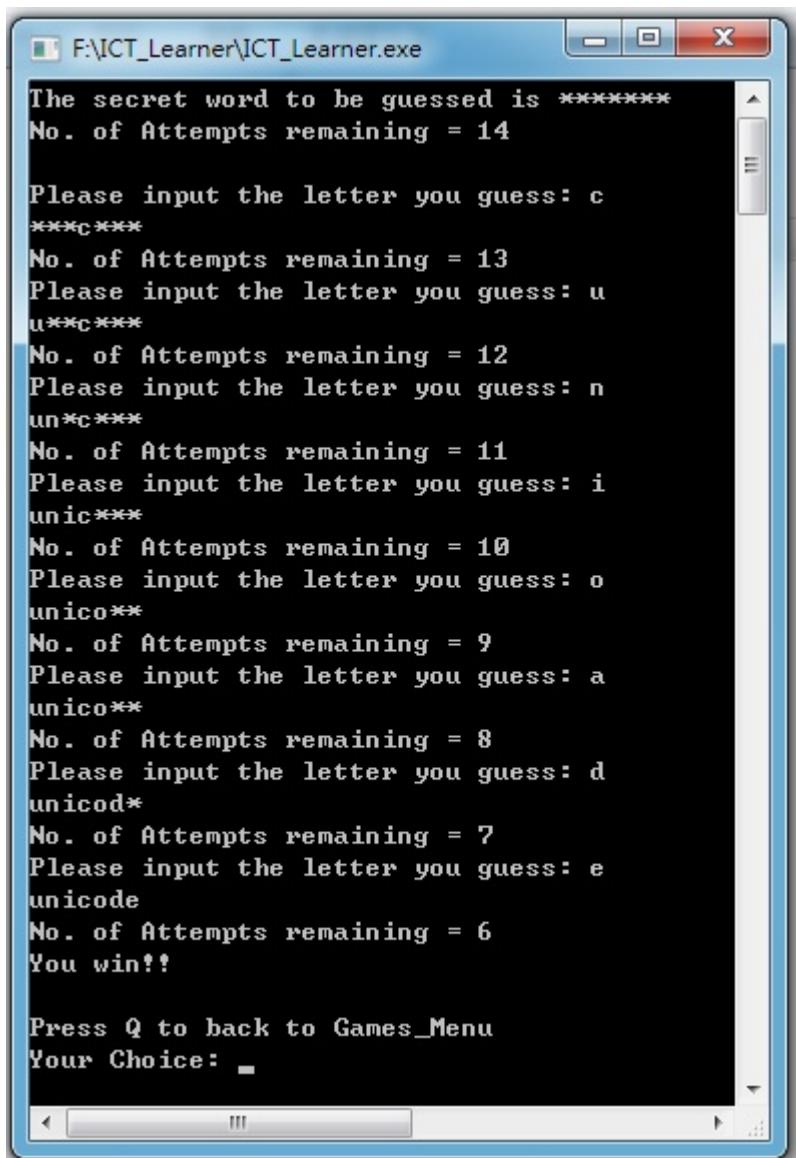
Vocabulary Guessing Game:

Menu:



This provides choice for users to make which to play.

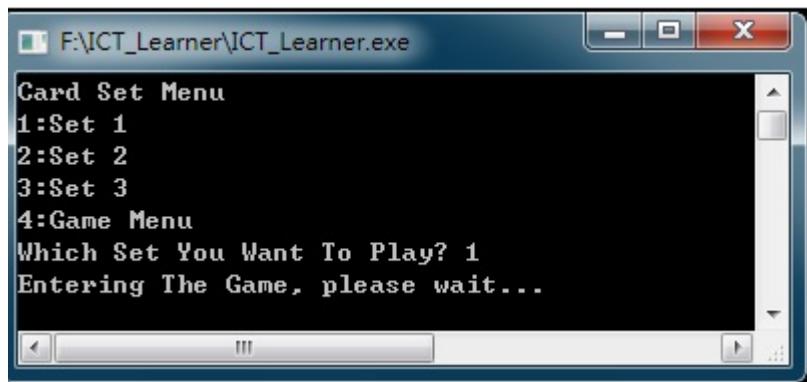
Playing:



In the game, users need to guess a word within a limited numbers of attempts, they will win the game if they get it, and otherwise they will get lost.

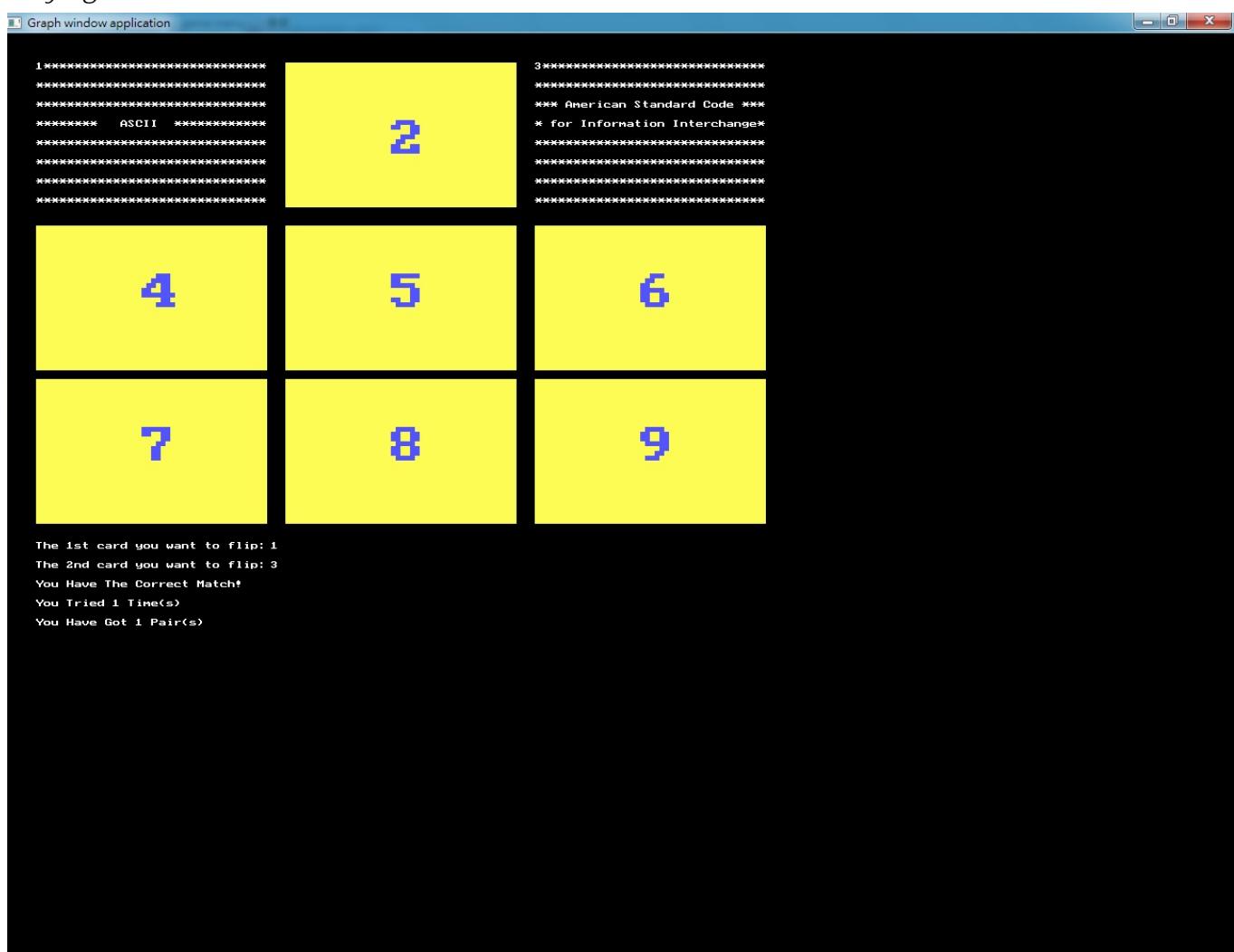
Matching Game:

Menu:



Same as the previous game, users can choose the card set to play ,and because opening graph take time, so there is a message to tell the users to wait.

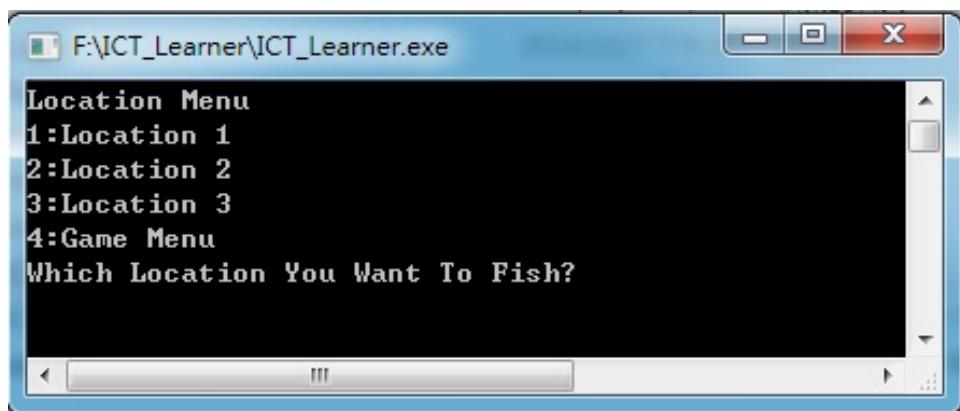
Playing:



The users need to flip 2 cards a time by pressing (1-9), the card remains flipped open if they get the correct pair, and otherwise the cards flip back in a few seconds later. There are no limited flipping times, therefore the game finished ONLY when the users got all pairs. The game will go directly back to the Main Menu after few seconds.

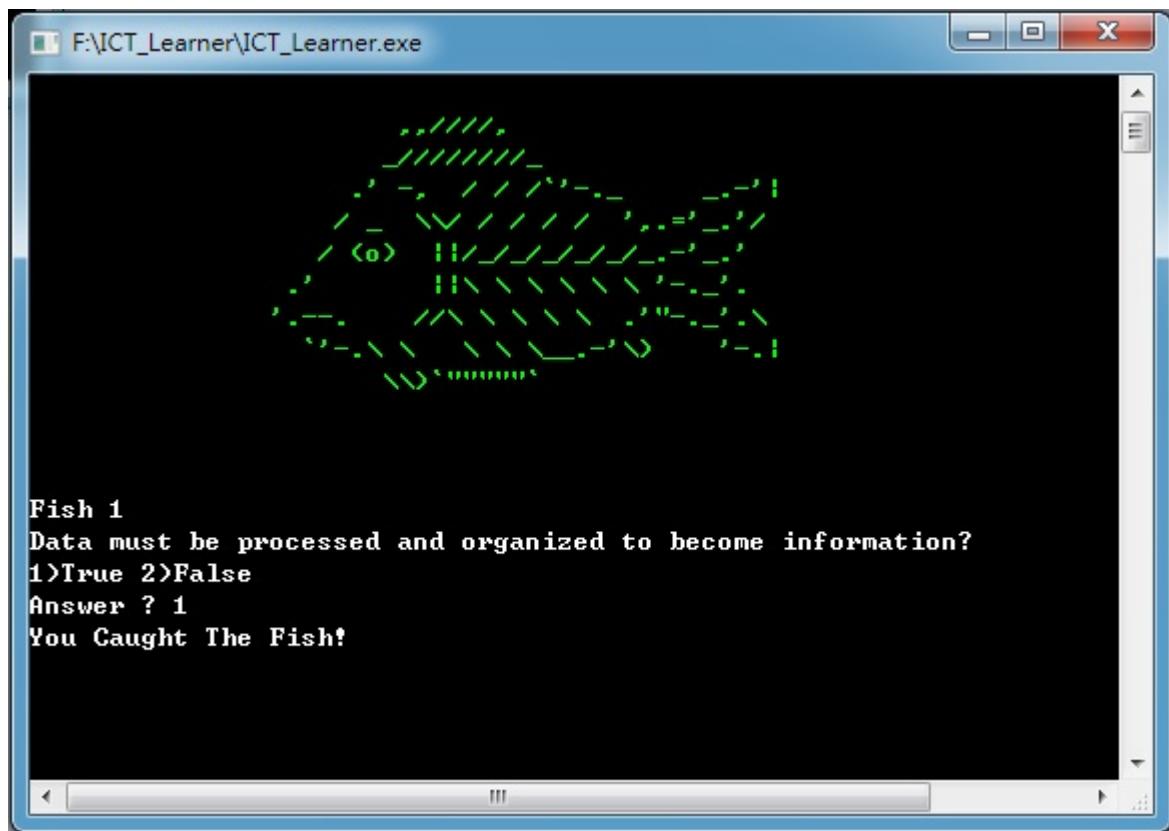
Fishing Game:

Menu:



Just the same. Users can choose which location to play.

Playing:

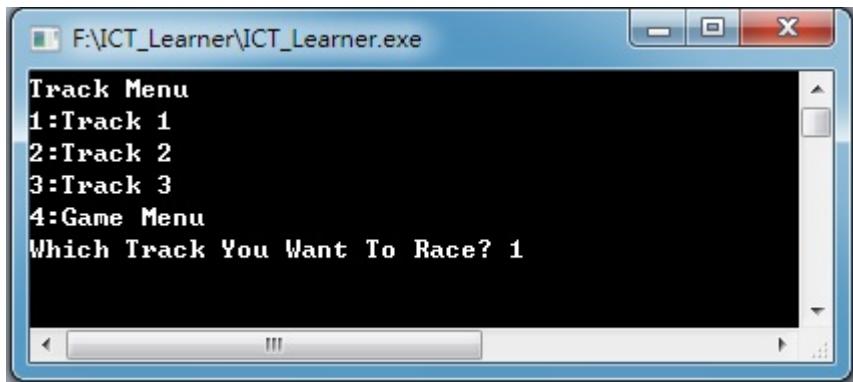


As shown in the picture, there will be a fish, the users need to ask a question in order to catch the fish. If they get the correct answer, they caught the fish, otherwise the fish get away. The color and the type of fish is randomly chosen in every question.

The format of the game is similar to the exercises.

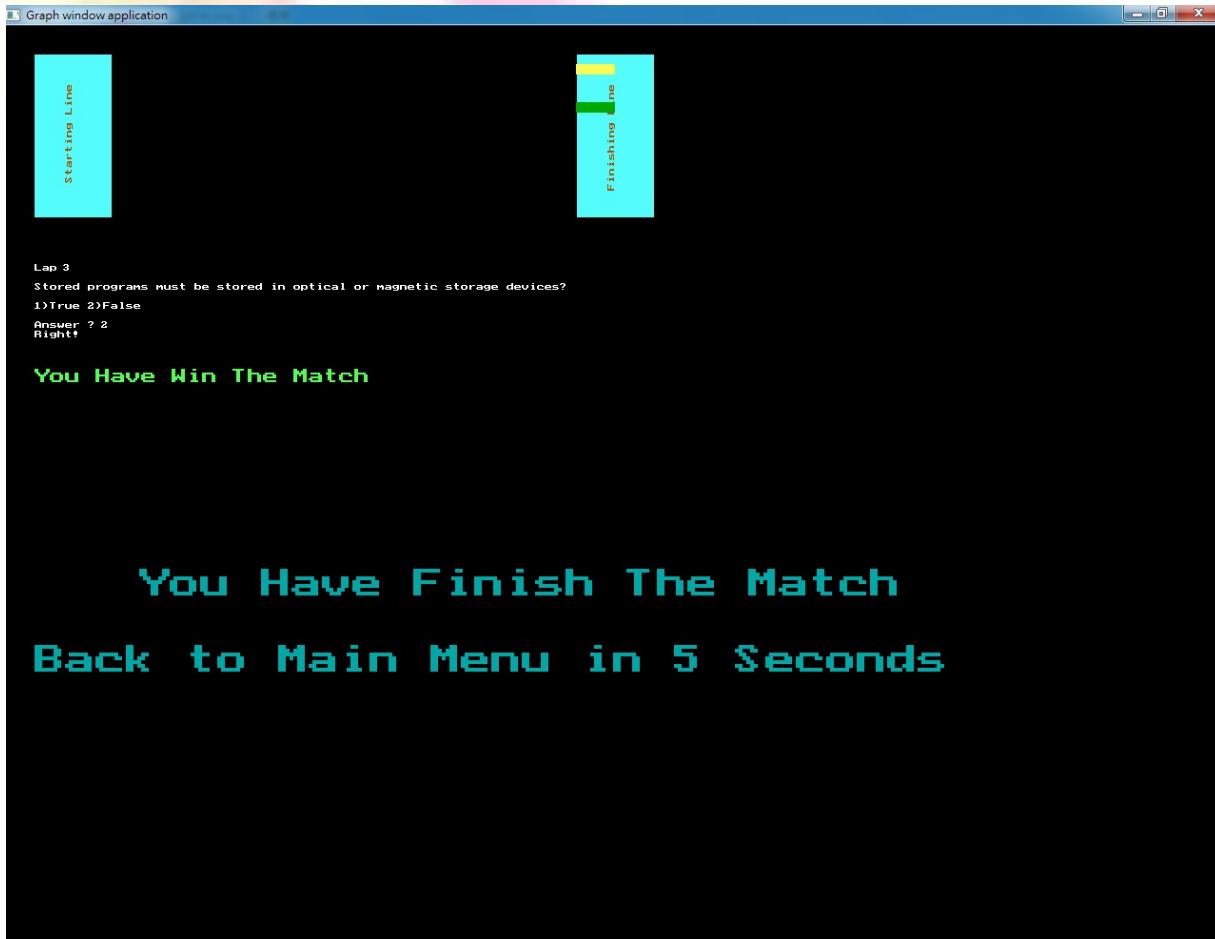
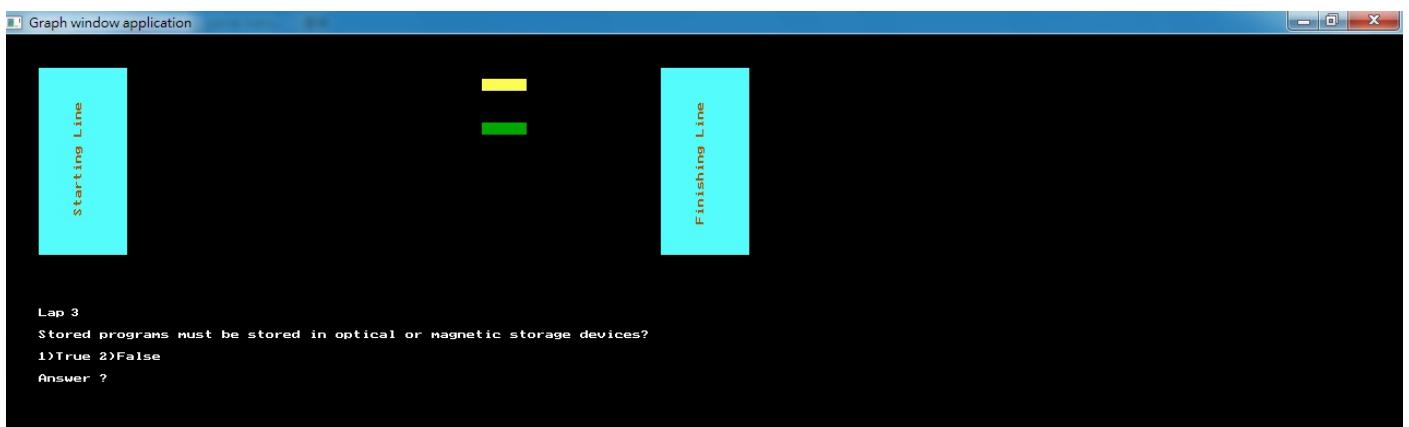
Car Racing Game:

Menu:



It should be dull to say again that, it is also the same. Just the users choose a track instead of a location.

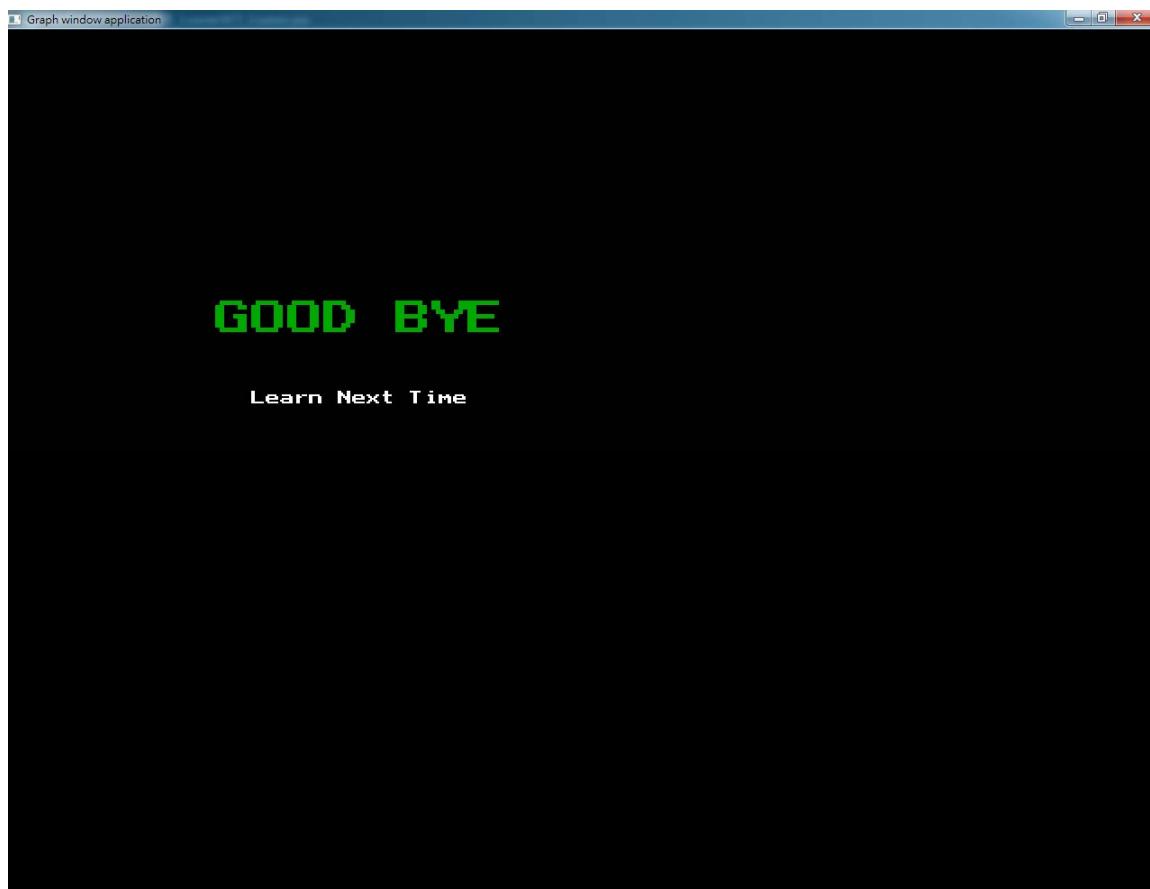
Playing:



The users answer question and move one step for every correct answer, the number of steps depends on numbers of questions. They can win the race if they get ALL questions CORRECT.

Good Bye Page:

79



Screenshots of files:

Login:

F4 ICT - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

Account	Password
Bobby Au	s401
Chan Kai Bong	s402
Chan Man Cheun	s403
Chan Mei Ling	s404
Tom Ho	s405
Lily Ho	s406

Here is the file format to store the accounts of students for them to login. The format of teachers accounts files and other classes are in same format.



The record file:

```

F4 ICT - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

Name
Quiz1 Quiz2 Quiz3
Bobby Au
-1 -1 -1
Chan Kai Bong
-1 -1 -1
Chan Man Cheun
-1 -1 -1
Chan Mei Ling
-1 -1 -1
Tom Ho
-1 -1 -1
Lily Ho
-1 -1 -1

```

Here is the file format of storing records of quizzes.

The resources:

Lists:

```

Note_C_List - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

Complusory Notes Menu
1.Chapter 1 Introduction to Information Processing (1.1-1.3)
2.Chapter 1 Introduction to Information Processing (1.4-1.5)
3.Chapter 1 Introduction to Information Processing (1.6-1.7)
4.Note 4
5.Note 5

```

It is a file to store what the resources are, all the same for notes, exercises and quizzes.

Notes

```

Note1 - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

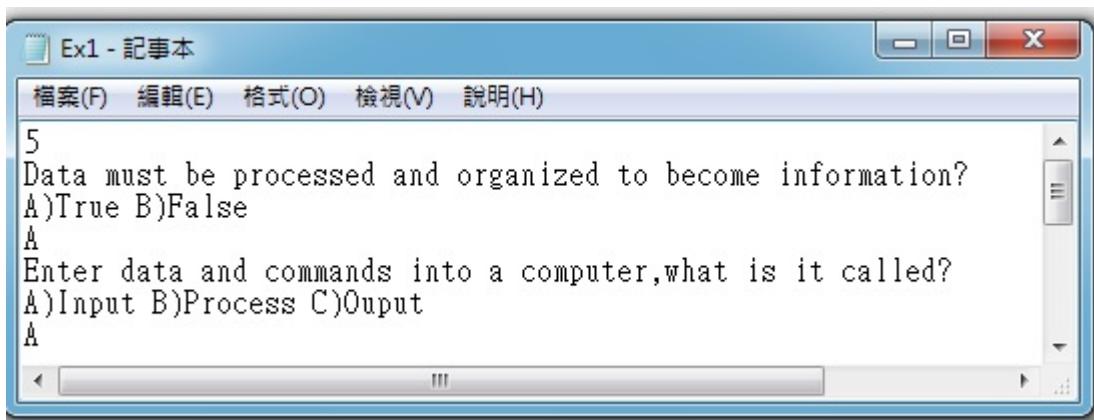
*****
* Chapter 1 Introduction to Information Processing *
*****

1.1 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 complicated calculations
  Produce the calculated results on different media
  Input-Process-Output cycle usually involves operations

```

It is the file storing the notes.

Doing questions:



This is a common file format in this program for storing questions. They are all the same format in exercises, quizzes, fishing game and car racing game.

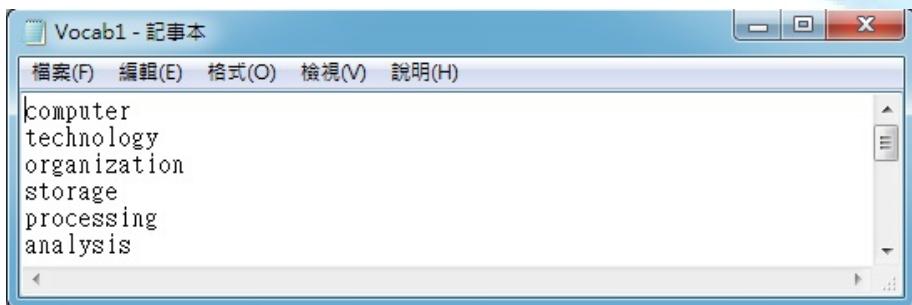
Comment page:



Just a statement to tell the users to type.

Special formats for games

Vocabulary game:



A list of vocabularies for the game to choose a word.

Fish file:



This is a file storing a fish, there are now 6 fishes in the aquarium (The folder storing fishes files. It is hoped that more fishes can be found for the game later.

***#After making the program, testing is a must for improving the program to make it to perform in this best condition.*

Chapter 5 Testing & Evaluation

5.1 Brief Description

In this chapter, I am going to:

1. Find out the bugs (logical and run-time errors) in the program
2. Check whether the program can achieve its purposes
3. See whether the program is user-friendly
4. Find out what information is needed to be given to the users in the user's manual
5. Debug and improve the program based on the testing and evaluation results
6. Discuss the pros and cons of the program and the ability of further development. (#Both the readability of the source code and the structure of the program)



5.2 Testing and Evaluation Plan

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

1. Internal testing and evaluation / Tested and evaluated by me (the programmer):
 - The program will be tested intensively by me – the programmer
 - I will prepare 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-friendliness, performance, flexibility for future development, reusability of program codes, etc.

2. External testing and evaluation / Tested and evaluated by users:
 - I will invite some (e.g. 10) targeted users/my classmates/friends to test and evaluate the program.
 - I will upload the object program (the .exe file) and some sample data files (a folder consist of all element) onto dropbox/google to let some users to try out the functions in their OWN computer to try the situation in local way (*Mainly test the usability and the functions)
 - I will upload the object program (the .exe file) and some sample data files into a shared folder onto my school e-class platform so that the users can try out cross users and computer situation.(Try out the acceptance of the updated time.)
 - I will upload the corresponding files to my blog / Web site and ask my friends to try
 - The users are invited to report the bugs they find and give their comments and suggestions on my program

#I will try to modify the program according to some improvement that I may want to add the reported bugs and suggestions. The new version of program after making modifications will be shown in Appendix 1.



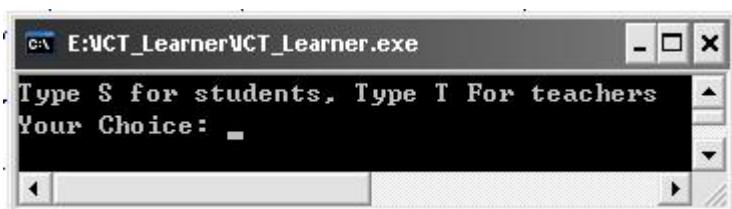
5.3 Internal Testing

- ❖ The test is done by the programmer (me)
- ❖ Different test cases are designed for each focus on different features of the program
- ❖ For each test case, I will
 - describe the purpose of the test case
 - describe the input
 - describe the expected output / results
 - show and describe the actual output / results. You may use screenshot(s)
 - state the conclusion of the test.
 - Suggest the modification(s) that should be made if there are unwanted results
 - Modify the program accordingly or explain why I am not able to modify it

*Before the testing, I found the color of the users' choice may not easy to see, most of them are changed to yellow

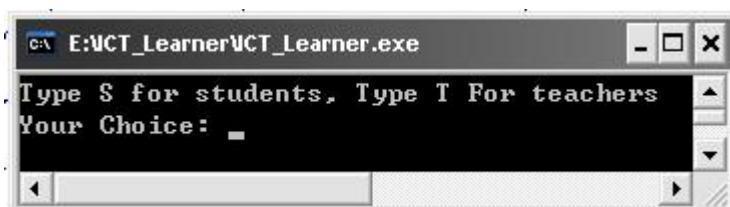
Test case 1 (Welcome Page)

Purpose:	To check the correctness of the output results on screen by entering the system and check if it get back to login procedure
Input:	Any key on the keyboard
Expected Output:	The Words shown in color and get back to the login procedure after typing a key.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

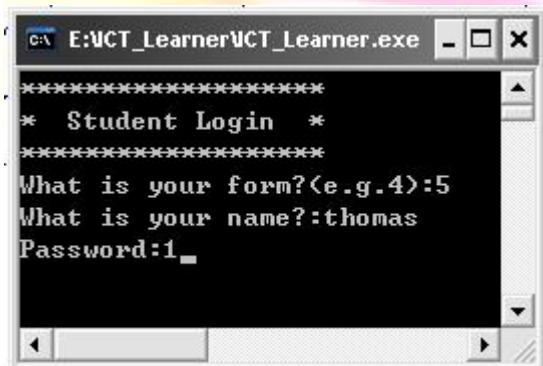


Test case 2 (Login Page)

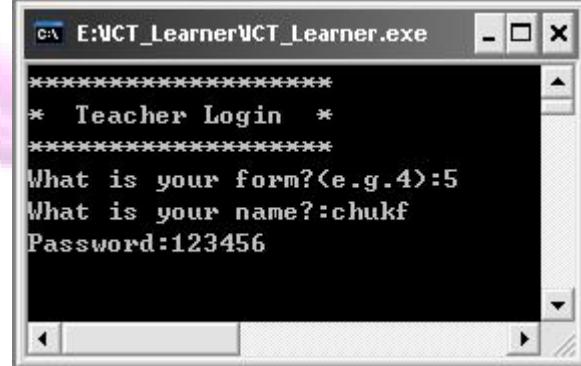
Purpose:	To check the correctness of the output results on screen by using normal input data
Input:	Normal account file and value All the information are inputted correctly during execution of program
Expected Output:	Login in to the correct login panel (teacher or student) Successfully get into menu
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	The choice of users make is changed to yellow.



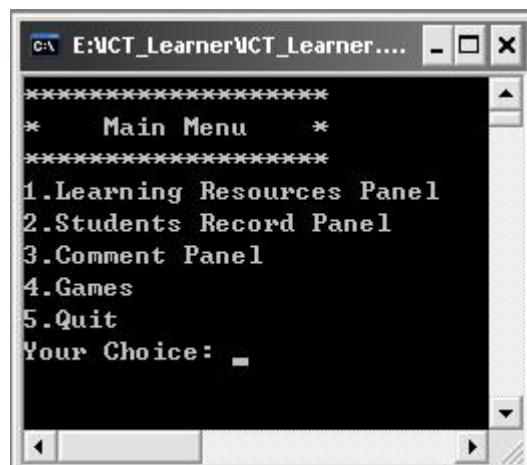
Type S on the above choice



Type T on the above choice



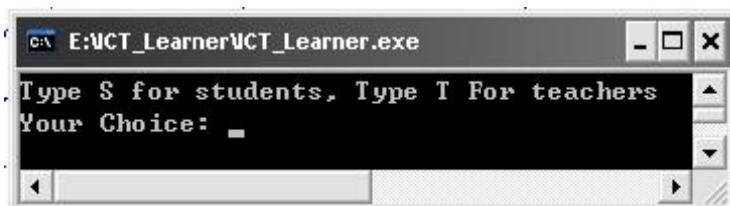
The information of the valid accounts is inputted.



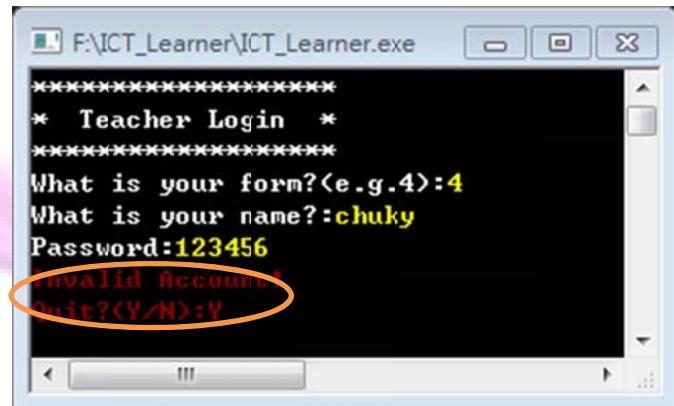
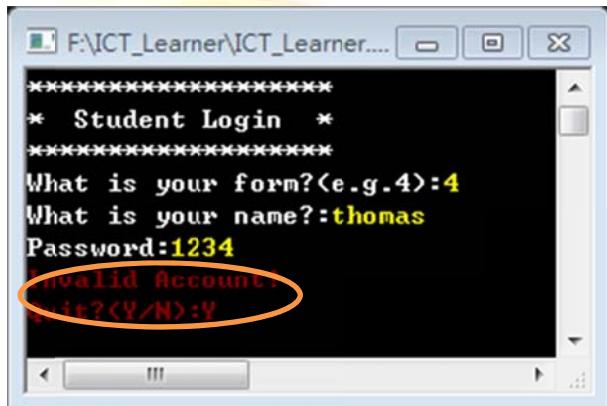
Both of them get to Main Menu !

Test case 3(Login Page2)

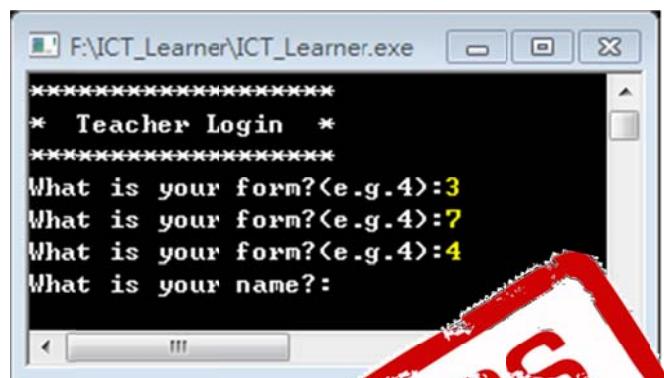
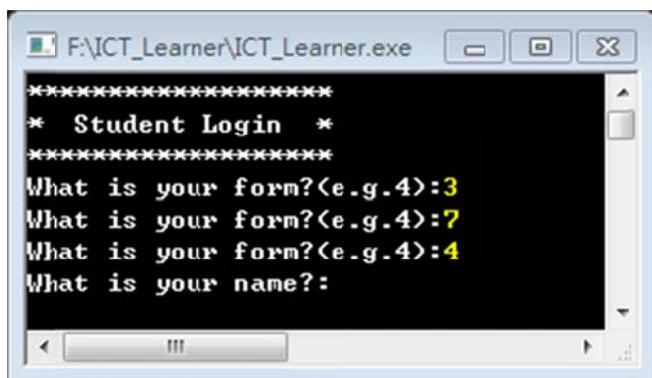
Purpose:	To test the program for the case when there are some invalid account information.
Input:	Normal account file An invalid account information
Expected Output:	It should ask the users if they leave or not.
Actual Output:	All actual results are the same as the expected results.
Test Result:	No bugs found
Follow-up Action:	Improve the part to accept 's' and 't'



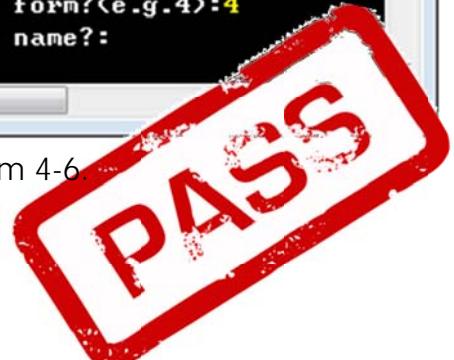
The system refresh and clear back to the same page if users type NOT 'S' or 'T'. However, to improve the usability the system is modify to accept 's' and 't'



Both Quit when type Y and get back to login for N.

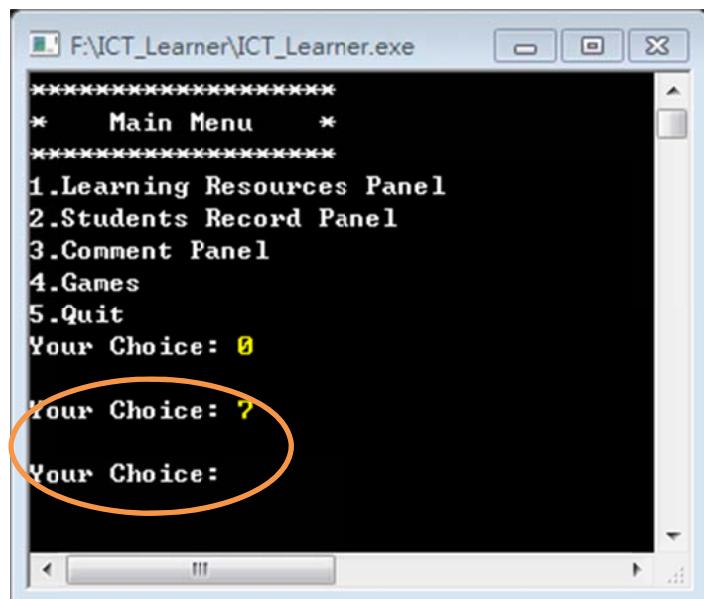


They ask the users to type again until the form is ranged from 4-6.



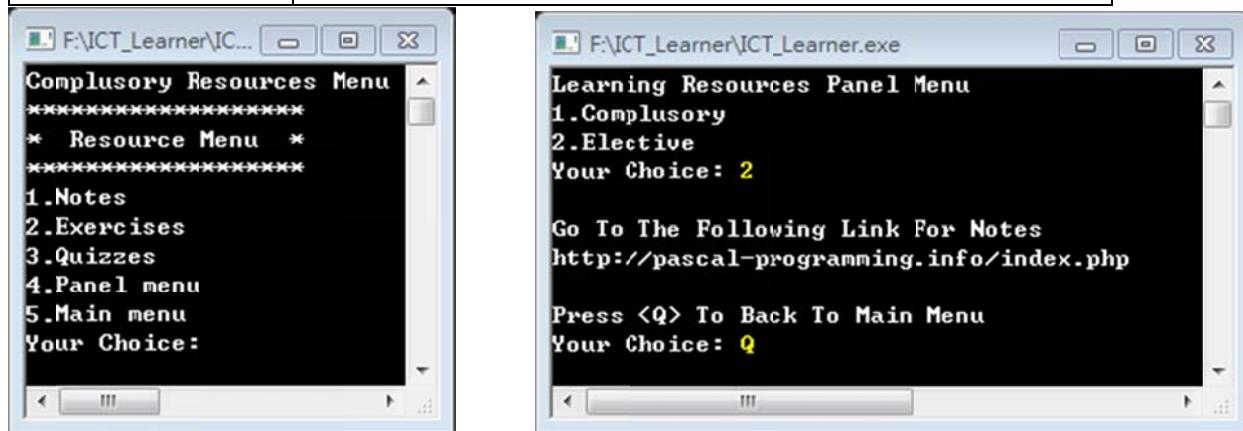
Test case 4 (Main menu)

Purpose:	To test the program for the case when there are wrong input of choice
Input:	A choice higher than 5 and a choice lower than 1
Expected Output:	A statement should tell the user to type again
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil

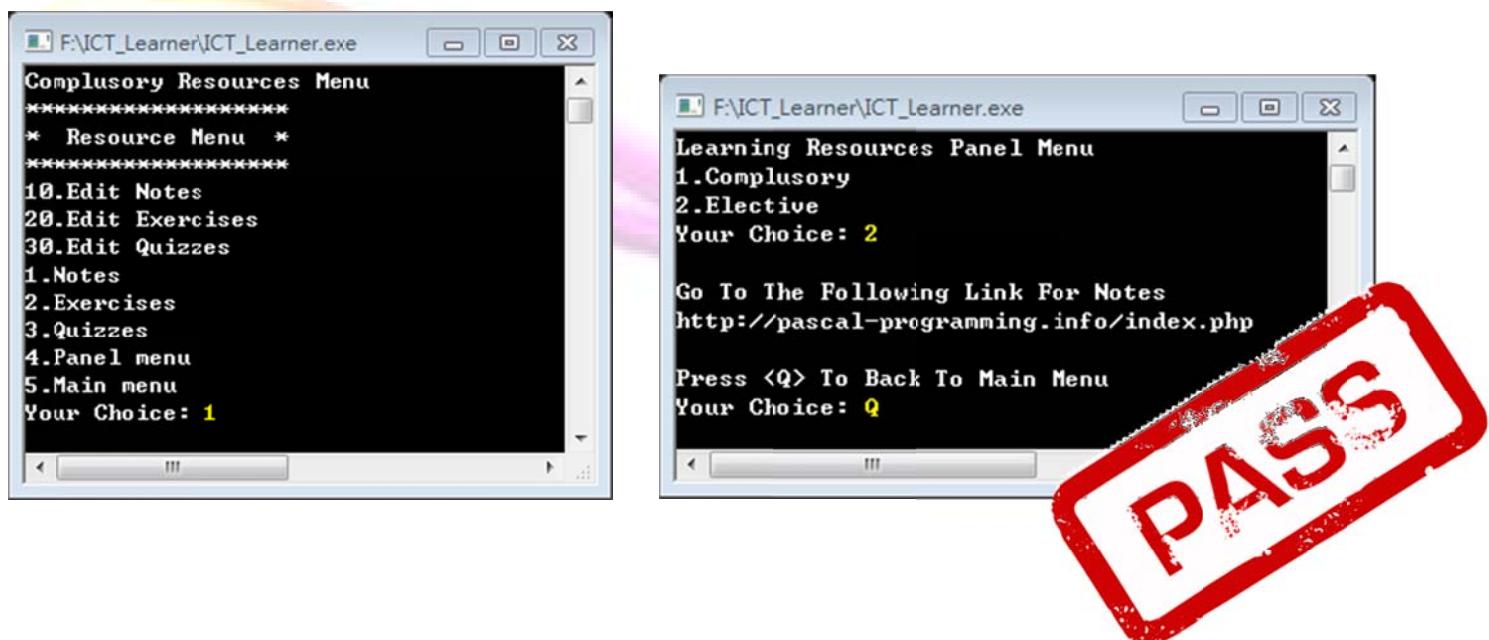


Test case 5 (Learning resource panel)

Purpose:	To test the program for the case with correct choice
Input:	Simple file of lists and choice
Expected Output:	Show the correct menu
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil



For the case of teacher



Test case 6 (Notes)

Purpose:	To test the program for the case with correct choice
Input:	Simple file of lists and content
Expected Output:	Show the correct menu and content
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil

```
F:\ICT_Learner\ICT_Learner.exe
Complusory Resources Menu
*****
* Resource Menu *
*****
10>Edit Notes
20>Edit Exercises
30>Edit Quizzes
1.Notes
2.Exercises
3.Quizzes
4.Panel menu
5.Main menu
Your Choice: 1

Complusory Notes Menu
1.Chapter 1 Introduction to Information Processing <1.1-1.3>
2.Chapter 1 Introduction to Information Processing <1.4-1.5>
3.Chapter 1 Introduction to Information Processing <1.6-1.7>
4.Note 4
5.Note 5
6:Resource Menu
Your Choice:
```

```
F:\ICT_Learner\ICT_Learner.exe
*****
* Chapter 1 Introduction to Information Processing *
*****

1.1 Basic Concepts of Input-Process-Output Cycle

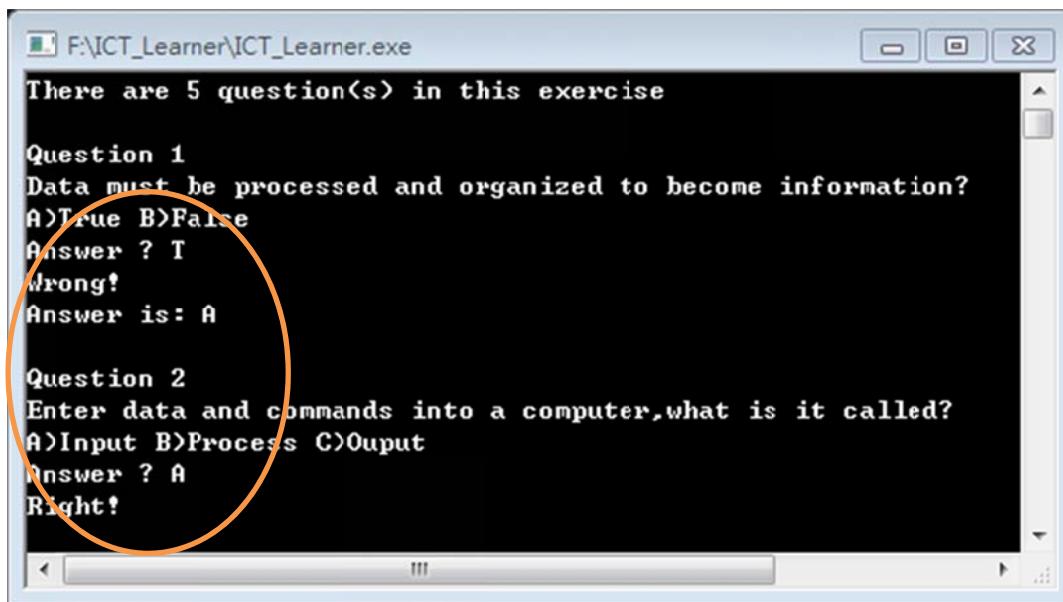
-A computer is a very powerful electronic machine
-A computer can:
  Store a huge amount of data
  Perform complicated calculations
  Produce the calculated results on different media
  Input-Process-Output cycle usually involves operations
  and hardware.
```



Test case 7 (Exercise)

90

Purpose:	To test the program for the case when the valid file of exercise and answer are used
Input:	Valid file with correct exercise format
Expected Output:	Show the questions, show answer when the user is wrong.
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	Nil

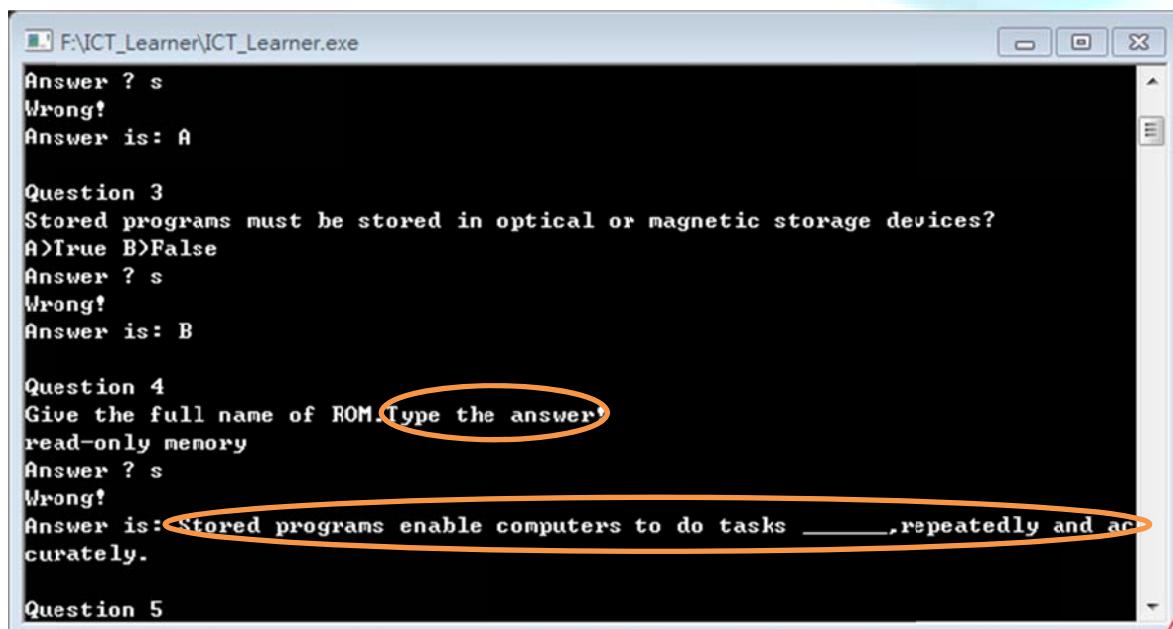
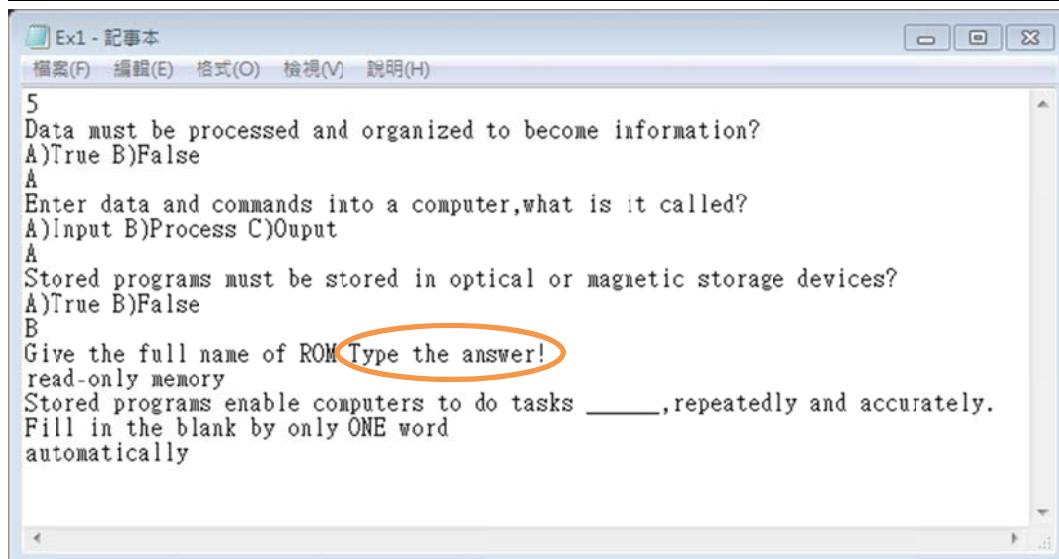


The system regards the wrong input or space as wrong.



Test case 8 (Exercise2)

Purpose:	To test the program for the case when the invalid file of exercise is used
Input:	Valid file with wrong exercise format
Expected Output:	There may be a runtime error
Actual Output:	NO statement is observed, the questions come out with wrong order and answers
Test Result:	Error
Follow-up Action:	It cannot easily solved by modifying the program, a guideline for teachers will added in the user manual



Test case 9 (Quiz and record)



Purpose:	To test the program for the case when the valid file of quiz and record file are used
Input:	Valid file with wrong exercise format
Expected Output:	The record will be updated
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL

Before doing quiz:

```
Name
Quiz1 Quiz2 Quiz3
Tony Au
-1 -1 -1
Donnie Chau
-1 -1 -1
Thomas Chung
-1 -1 -1
Melanie Fung
-1 -1 -1
Alex Kwok
-1 -1 -1
```

Quiz:

```
Question 4
Give the full name of ROM.
Type the answer!
Answer ? read-only memory
Right!

Question 5
Give the full name of ASCII
Type your answer
Answer ? .
Correct

You have 2 mark(s)
Answer will be given by teachers

Press Q to back to Quizzes List
Your Choice:
```

After doing quiz:

```
Name
Quiz1 Quiz2 Quiz3
Tony Au
-1 -1 -1
Donnie Chau
-1 -1 -1
Thomas Chung
2 -1 -1
Melanie Fung
-1 -1 -1
Alex Kwok
-1 -1 -1
```

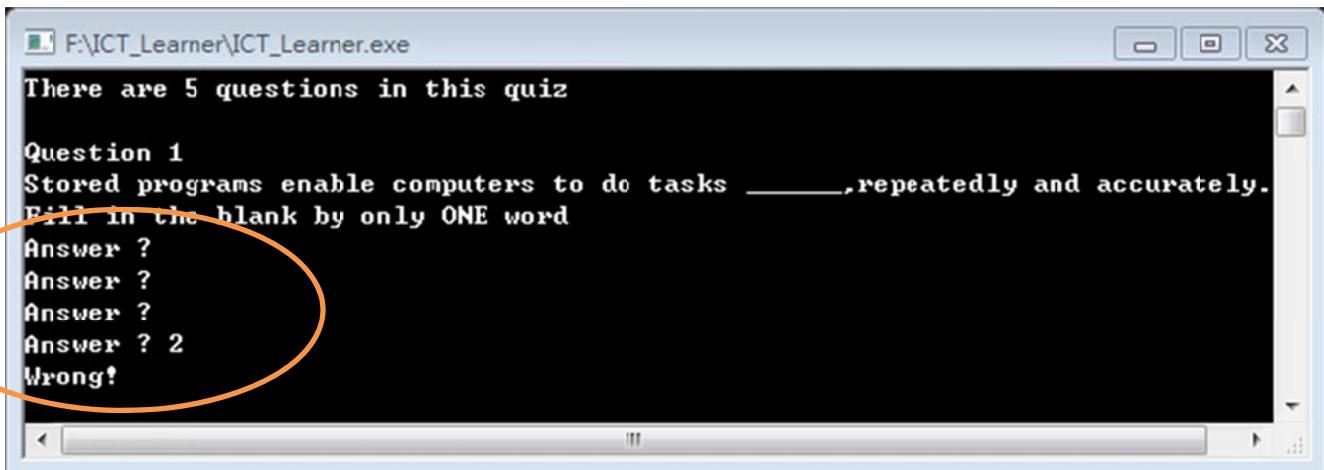
The record is updated!



Test case 10 (Quiz2)

Purpose:	To test the program for the case when the invalid
----------	---

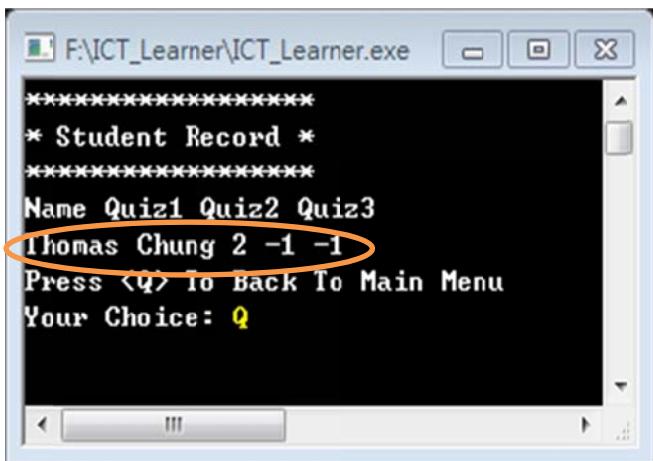
	input data
Input:	Valid file with wrong input answer
Expected Output:	Type again for space and the wrong input is regarded as wrong answer.
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL



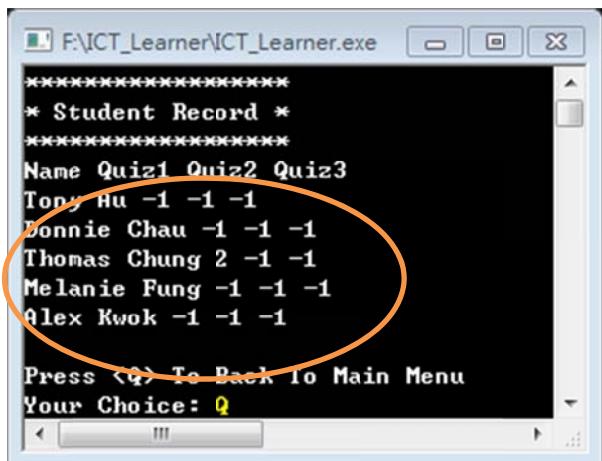
Test case 11 (Record panel)

Purpose:	To test the program for the case when the valid input data
Input:	Valid file with correct record format
Expected Output:	Show the own record for students Show list of records of students for teachers
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL

Students:



Teachers:

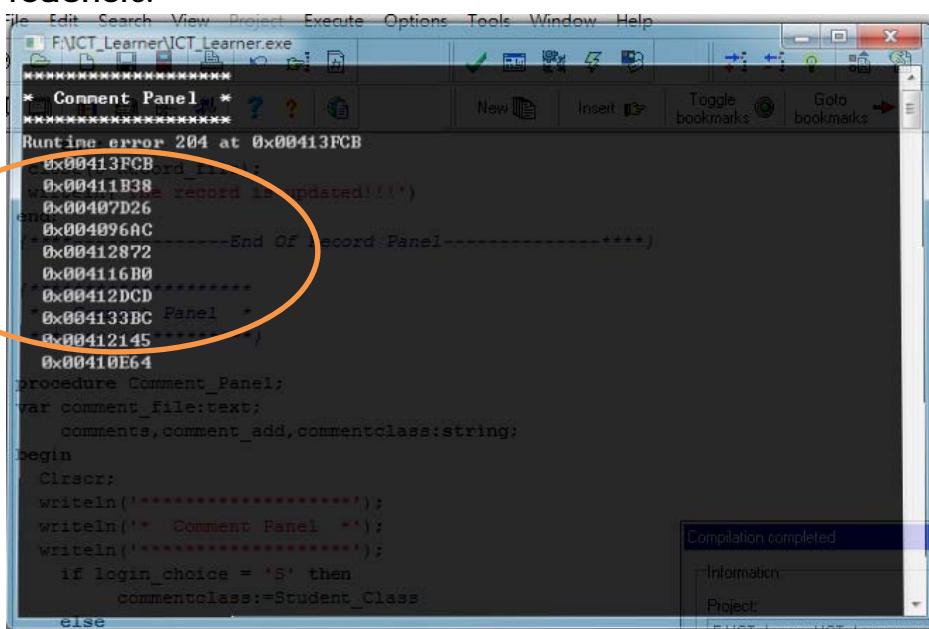


Test case 12 (Comment Panel)

Purpose:	To test the program for the case with correct data
----------	--

	files
Input:	Valid file with correct input answer
Expected Output:	Show the comments
Actual Output:	Same as expected for students , runtime error for teachers
Test Result:	Teachers cannot read the file
Follow-up Action:	NIL

Teachers:



The screenshot shows a software window titled "F:\ICT_Learner\ICT_Learner.exe". The main area displays a runtime error message:

```

*****
* Comment Panel *
*****
Runtime error 204 at 0x00413FCB
0x00413FCB
0x00411B38 record is updated!!!!)
0x00407D26
0x004096AC
-----End Of Record Panel-----
0x00412872
0x004116B0
0x00412DCD
0x004133BC Panel +
0x00412145 ****)
0x00410E64

procedure Comment_Panel;
var comment_file:text;
  comments,comment_add,commentclass:string;
begin
  Clrscr;
  writeln('*****');
  writeln('* Comment Panel *');
  writeln('*****');
  if login_choice = 'S' then
    commentclass:=Student_Class
  else

```

A red circle highlights the error code `0x00413FCB`. A small window titled "Information" is visible in the bottom right corner, showing "Compilation completed".



After modify:



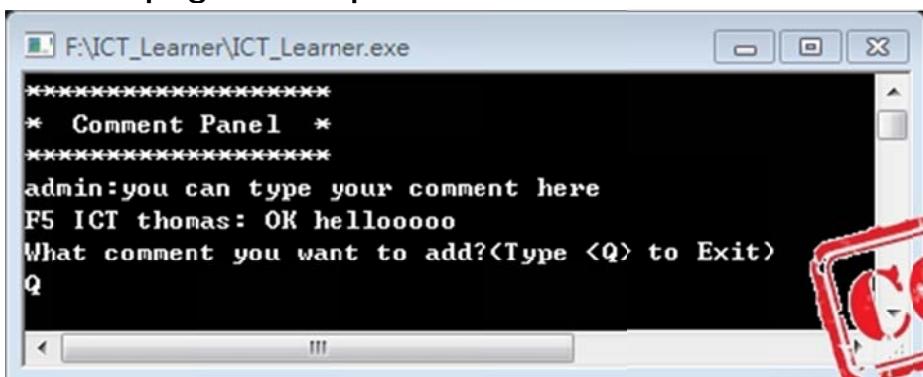
The screenshot shows the same software window after modification. The output now displays user input and a question:

```

*****
* Comment Panel *
*****
admin:you can type your comment here
F5 ICT thomas: OK hellooooo
What comment you want to add?(Type <Q> to Exit)
Q

```

Students page has no problem:



The screenshot shows the software running successfully for students. The output is identical to the modified version above:

```

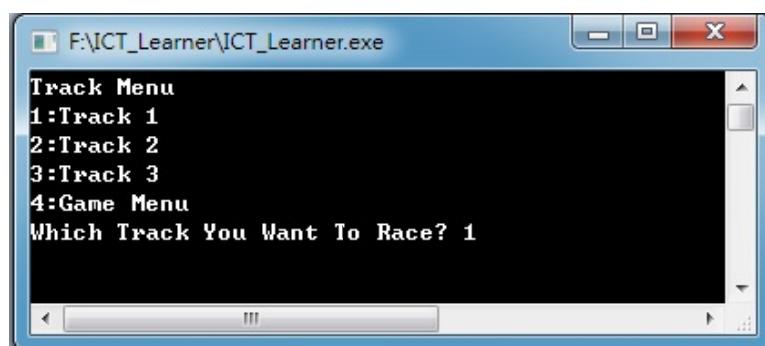
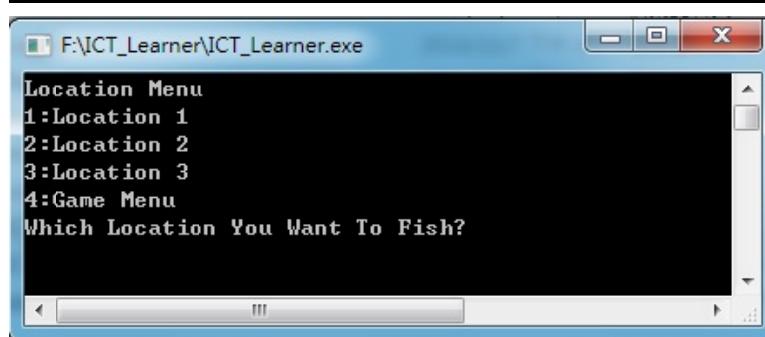
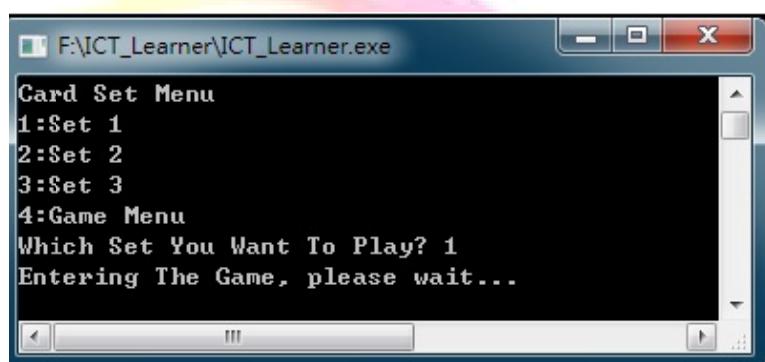
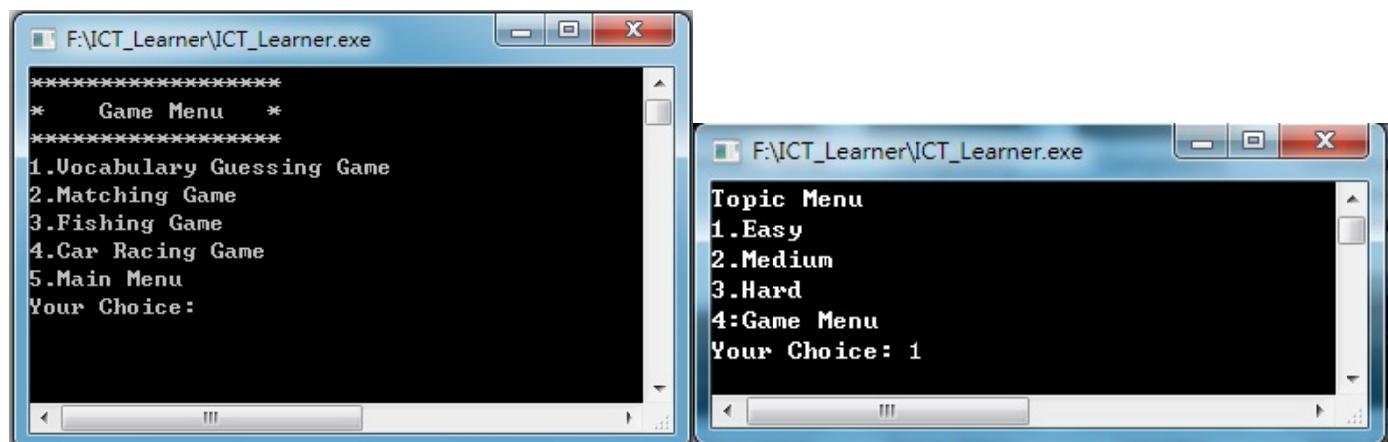
*****
* Comment Panel *
*****
admin:you can type your comment here
F5 ICT thomas: OK hellooooo
What comment you want to add?(Type <Q> to Exit)
Q

```



Test case 13 (Games Menu)

Purpose:	To test the program for the part of games
Input:	Valid file with correct input answer
Expected Output:	Show the game menus
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL



PASS

Test case 14 (Vocabulary Guessing Game)

Purpose:	To test the program for the case with valid data and check if it gain another word next time
Input:	Valid file with correct input answer
Expected Output:	Show the game answer after the remains becomes zero or users get correct.
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL

1st:

```
b*tmap
No. of Attempts remaining = ?
Please input the letter you guess: i
bitmap
No. of Attempts remaining = 6
You win!!
Press Q to back to Games_Menu
Your Choice:
```

2nd:

```
Please input the letter you guess: r
computer
No. of Attempts remaining = 6
You win!!
Press Q to back to Games_Menu
Your Choice:
```



```
The secret word to be guessed is *****
No. of Attempts remaining = 16

Please input the letter you guess: c
*****
No. of Attempts remaining = 15
Please input the letter you guess: fg
*****
No. of Attempts remaining = 14
Please input the letter you guess: om
*****
```

If the user type in two letters, even he gets 2 words correct, only one will be accepted.

Test case 15 (Vocabulary Guessing Game 2)

Purpose:	To test the program for the case with valid data and check it with incorrect data.
Input:	Valid file with incorrect input answer or duplicated answer
Expected Output:	Show the input is wrong
Actual Output:	No response for duplicated letter and not a letter
Test Result:	There are some inconvenience for users
Follow-up Action:	A better interface is made and the errors are fixed

```
G:\ICT_Learner\ICT_Learner.exe
The secret word to be guessed is *****
No. of Attempts remaining = 20

Please input the letter you guess: s
*****
No. of Attempts remaining = 19
Please input the letter you guess: s
*****
No. of Attempts remaining = 18
Please input the letter you guess: 1
*****
No. of Attempts remaining = 17
Please input the letter you guess:
```

After typing the first letter, the interface below is shown.

The letter used is shown to users and the previous statements are clear to make a clear screen.

```
F:\ICT_Learner\ICT_Learner.exe
The secret word: *****s****s****
No. of Attempts remaining = 23
The letter used: s
Please input the letter you guess(small letter):
```

Type 's' again:

```
F:\ICT_Learner\ICT_Learner.exe
This letter has been chosen!!!
The secret word: *****s****s****
No. of Attempts remaining = 23
The letter used: s
Please input the letter you guess(small letter):
```

Type the words that are not (a-z):

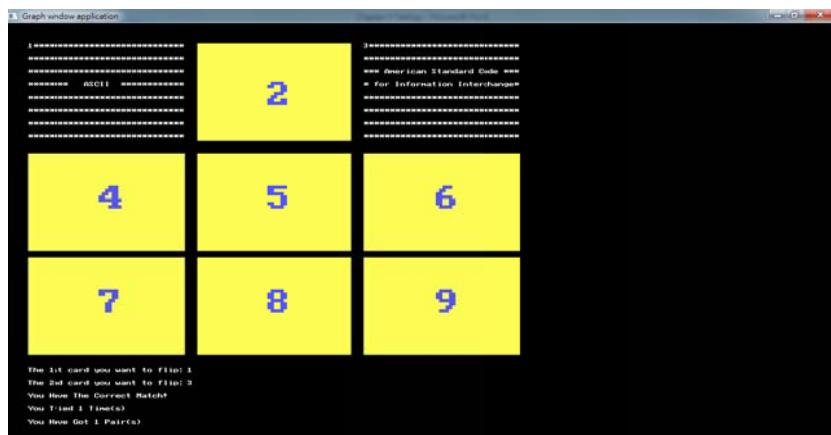
```
F:\ICT_Learner\ICT Learner.exe
Please Enter A Letter(a-z)!!!
The secret word: *****s****s****
No. of Attempts remaining = 23
The letter used: s
Please input the letter you guess(small letter):
```

CORRECTED

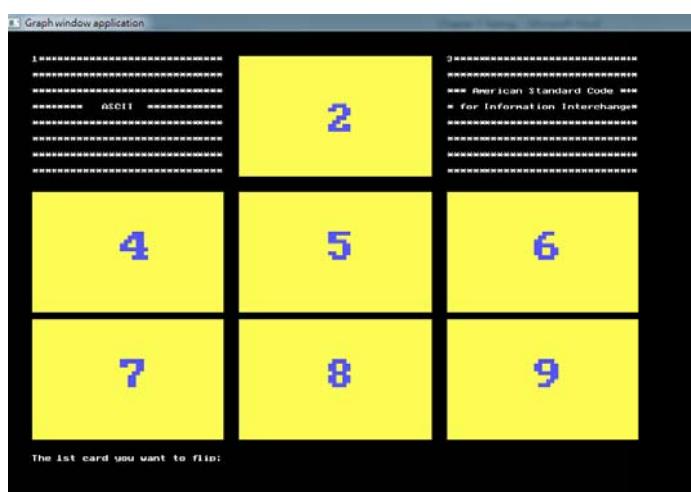
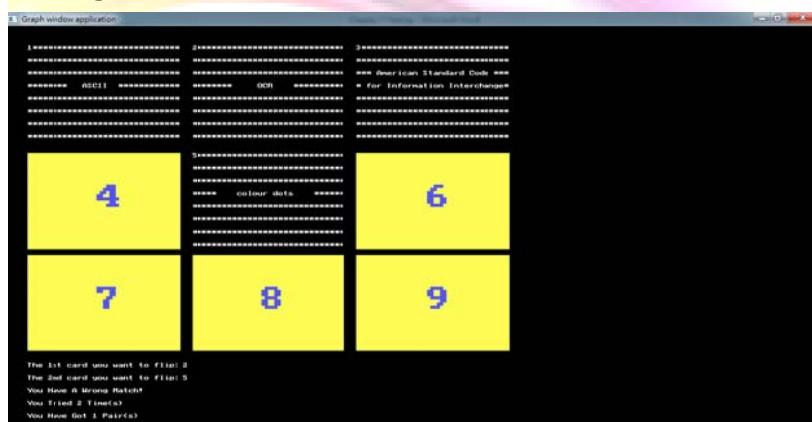
Test case 16 (Matching Game)

Purpose:	To test the program for the case with valid data with valid input
Input:	Valid file with correct input answer
Expected Output:	Show as wrong when wrong pair made The cards still flipped after correct
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL

Correct:



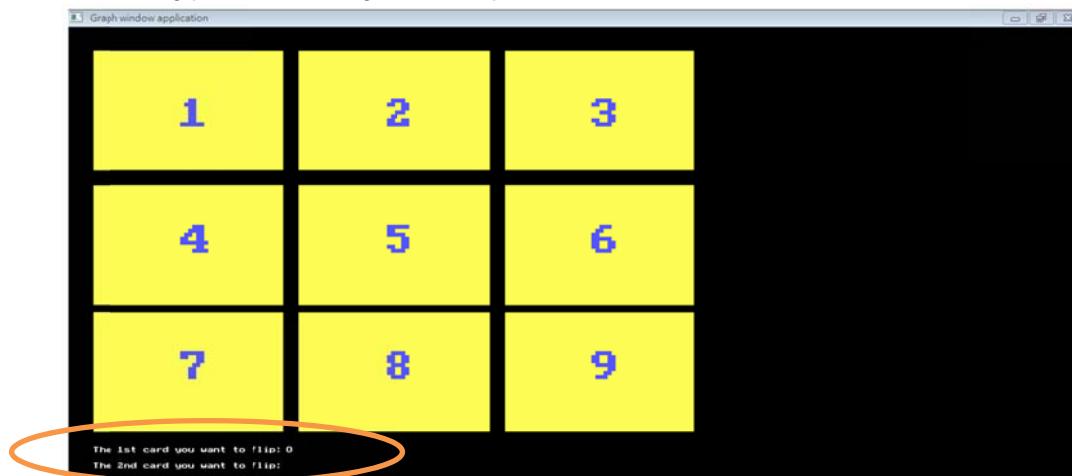
Wrong:



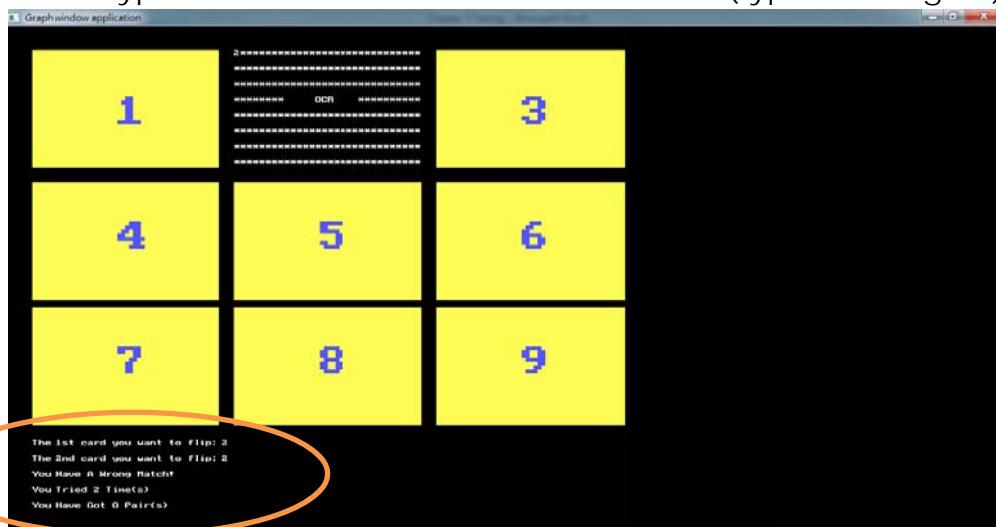
Test case 17 (Matching Game2)

Purpose:	To test the program for the case with valid data with invalid input
Input:	invalid file with correct input answer
Expected Output:	Show as wrong
Actual Output:	Same as expected
Test Result:	However, regard as wrong when typing the choices which are not shown and the duplicated choices may bore the users. In view of this, I did some changes such that it can be more user-friendly.
Follow-up Action:	Correct the program to make responses

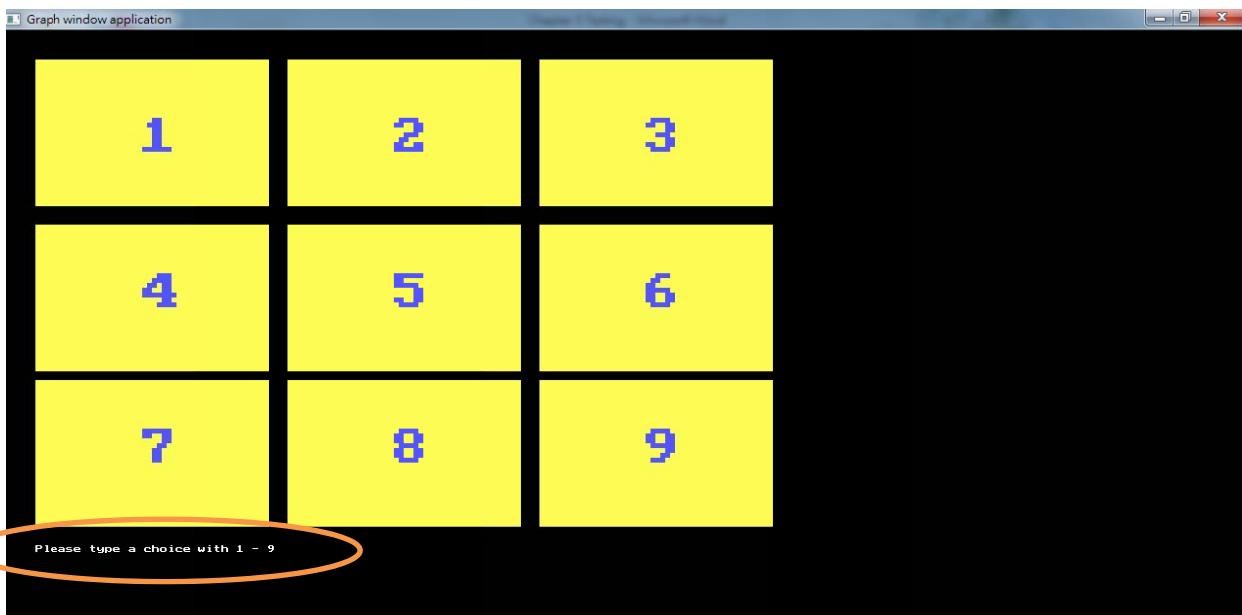
When type in '0' key, no response is made (No that card):



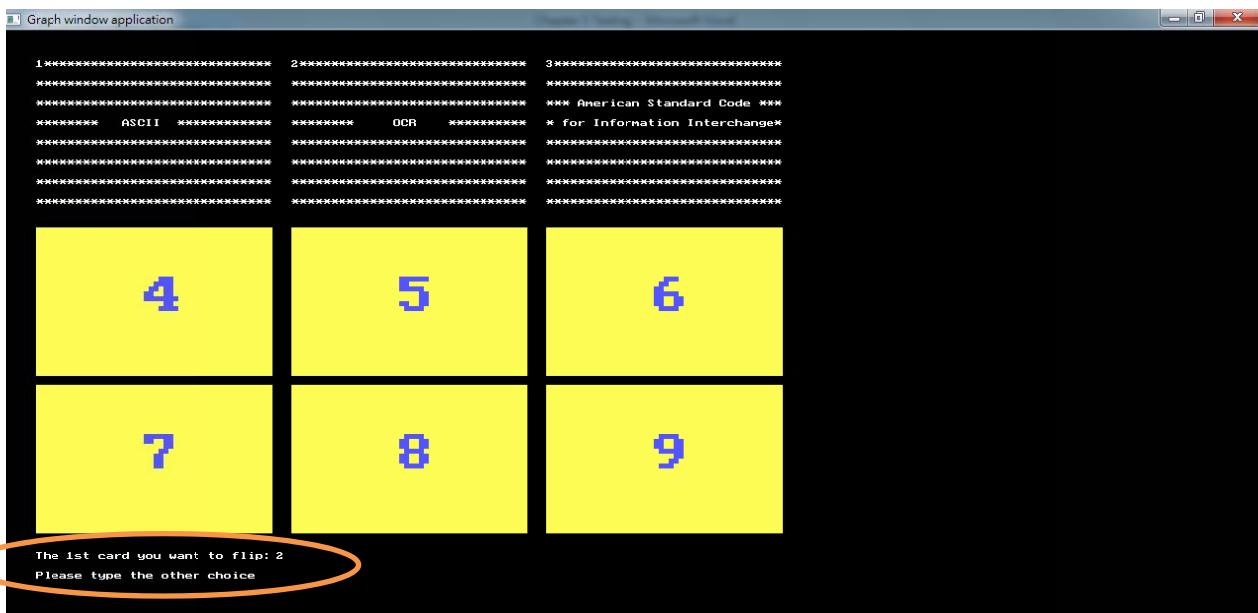
When type in the same choice with 1st choice (type in '2' again):



When type in '0' key, a response is made (No that card):



When type in the same choice with 1st choice(type in '2' again):



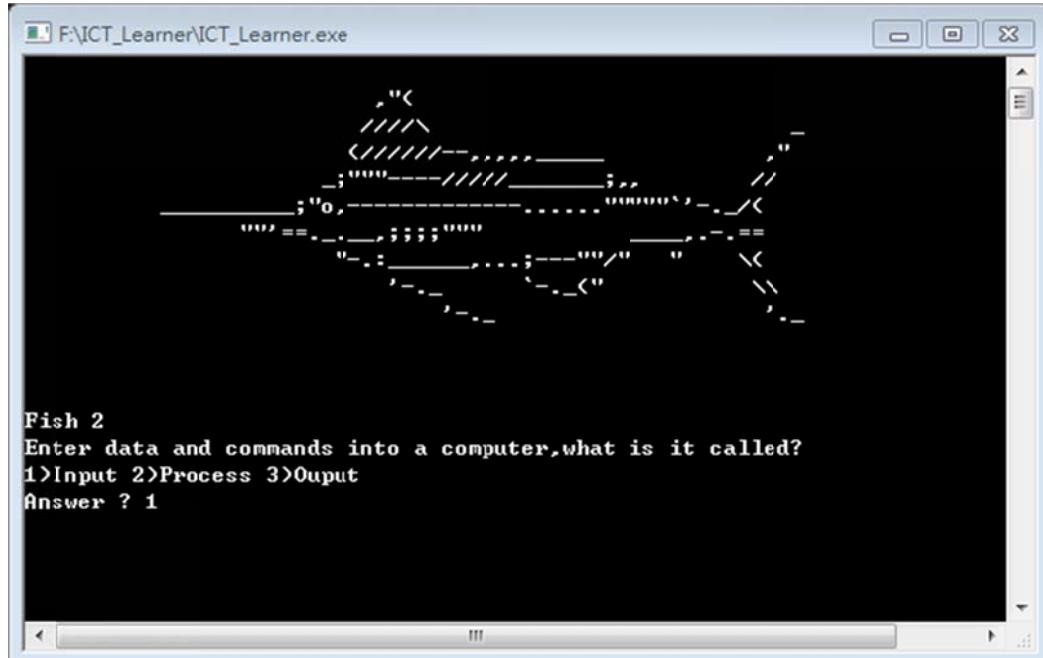
It makes the game easier to play and it can have more fun.

can have more fun.

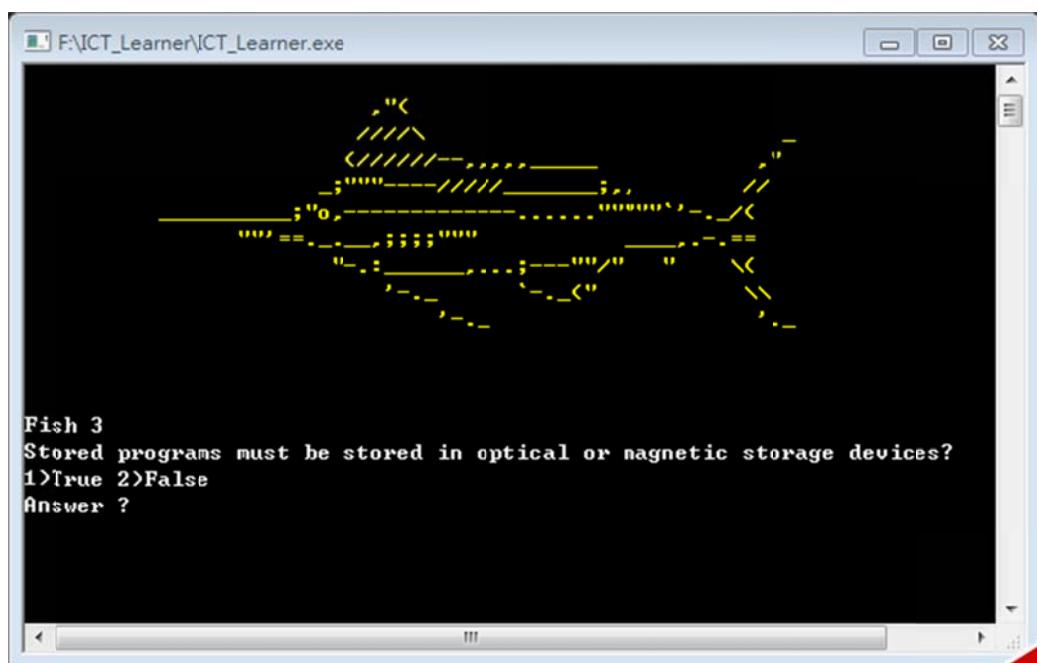
Test case 18 (Fishing Game)

Purpose:	To test the program for the case with valid data with invalid input, test if the fish will change in every question
Input:	Valid file with incorrect input answer
Expected Output:	It will regard as the wrong answer Fish change in every question
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL

Fish2:



Fish3:



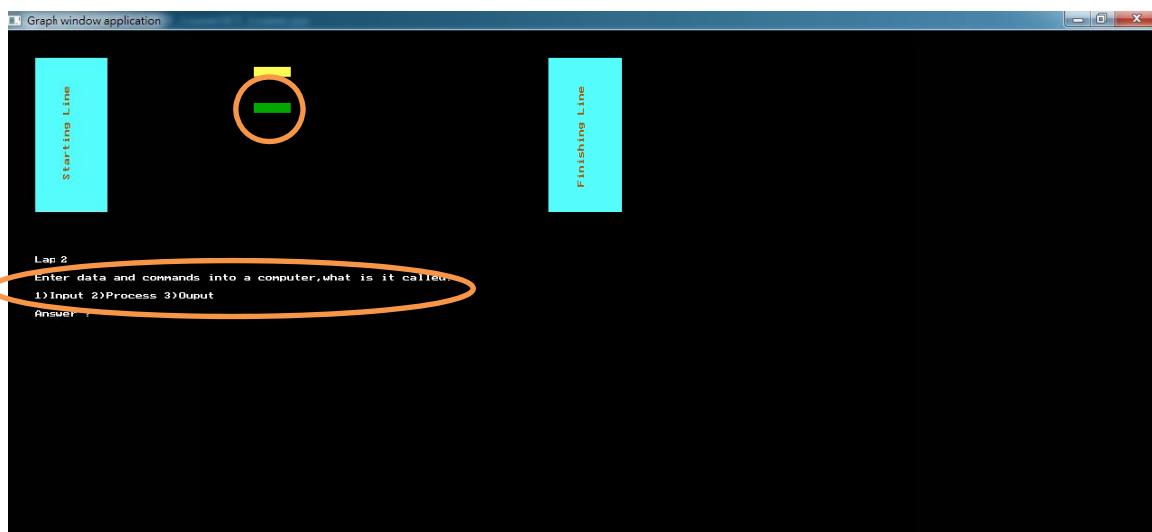
#Color is changed, it is called the other fish

Test case 19 (Car Racing Game)

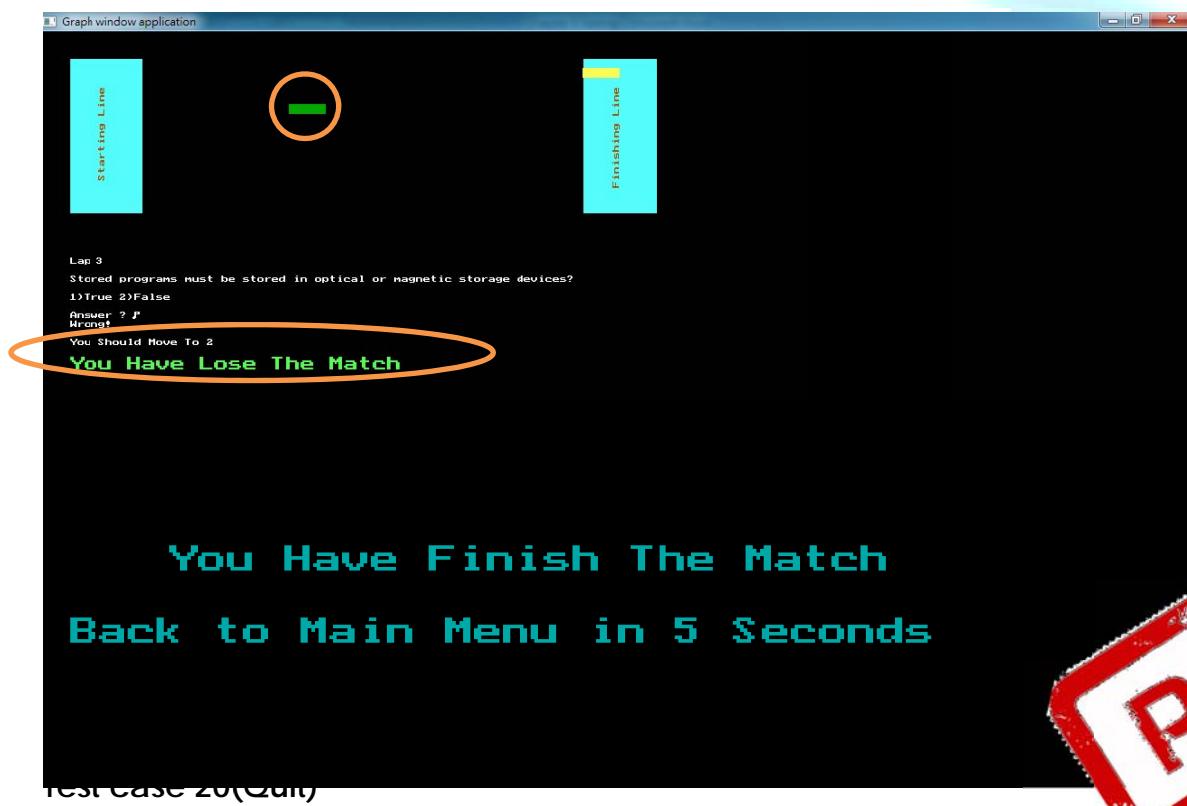


Purpose:	To test the program for the case with valid data with invalid input and valid input
Input:	Valid file with incorrect input answer and correct answer
Expected Output:	It will regard as the wrong answer Car move with correct answer ONLY If not answer all the question correctly, it regards as lose
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL

1st answer correct:



2nd answer wrong:



Purpose:	To test the program for the case with valid data
Input:	Valid file
Expected Output:	Show a statement of updating file and show good bye page.
Actual Output:	Same as expected
Test Result:	Pass
Follow-up Action:	NIL

```

Dev-Pascal 1.9.2 - FVCT_Learner.pas
File Edit View Insert Options Tools Help
F:\VCT\Learnervct\Learnervct.exe
=====
* Main Menu *
=====
1.Learning Resources Panel
2.Students Record Panel
3.Comment Panel
4.Games
5.Quit
Your Selection: 3
The record is updated!!!
GOOD BYE
Learn Next Time
=====
commentclass;=Teacher_Class* INT';
Assign(comment_file,HDD+'\ICT_Learner\Comment\'+commentclass);
reset(comment_file);
while not eof(comment_file) do
begin
readln(comment_file,comments);
writeln(comments);
end;
Close(comment_file);
Assign(comment_file,HDD+'\ICT_Learner\Comment\'+commentclass);
Append(comment_file);
writeln(comment_file,commentclass+' '+c_name+': '+ comment_add);
else
if login_choice = 'S' then
Students_Choices_List
else
Teachers_Choices_List;
=====
45893   Insertion 1456 lines in file
Compiler Resource Compile log
List... Unit Message
F:\VCT\Learnervct\Learnervct.pas compiled successfully

```



5.4 Self-Evaluation

- Good features of the program:
 - ❖ Functions
 - There are enough functions for the basic learning for students
 - It reduces the use of paper and time to give out notes, exercises and quiz.
 - The game panel can help student to relax and learn.
 - The teachers can check the learning process of the students
 - Users can share their comments and difficulties to discuss.
 - ❖ User-friendless
 - Every panels and procedure are well planned
 - Users can easily use it by simple texting
 - Not much equipment is needed.
 - There are colorful scene to attract the users
 - ❖ Performance
 - The performance is satisfactory.
 - The response time is acceptable.
 - ❖ Flexibility for future development
 - It is easy to add resources to the folder
 - An elective panel is ready for further development
 - ❖ Reusability of program codes
 - Codes in the *.pas file are classify for different parts
 - The next programmer can read the code easily
 - The codes are not made for one specific case.
- Shortcomings of the program:
 - ❖ Functions
 - There are so many functions that the users may need time to adapt to it.
 - It is not easy to take over the black and white
 - Classes are still needed to conduct by teacher
 - The quizzes need to trust the students that they won't copy answers.
 - ❖ User-friendless
 - The design of the interface may be too dull.
 - Teacher play an important role in using this learner
 - ❖ Performance
 - Some parts are not too perfect
 - ❖ Flexibility for future development
 - There are limitation in the number of lines in the files
 - There are standard file format and structure for program.
 - ❖ Reusability of program codes
 - The procedures call through each other, it may not easily checked by the next programmer for the flow.

5.5 External Testing and Evaluation

The program is uploaded online for the users to give advices.

A survey is conducted for the results.

Some follow up will be done for the improvement.

About 15 students test the program internally with their computers.

About 4 teachers test the program with their computer.

And the program is further reviewed by an experienced reviewer and programmer.

Some inconveniences are found and follow up are made.

The screenshot shows a Dropbox interface with a blue diamond logo and the word "Dropbox". On the left, there's a sidebar with icons for "文件" (Files), "照片" (Photos), "共享" (Sharing), "链接" (Links), "事件" (Events), and "开始使用" (Get Started). The main area displays a list of files:

	名称	类型	最后修改时间
	Yahoo Mail	文件夹	--
	Yahoo! Mail	文件夹	--
	Dropbox 快速入门.pdf	文档	5/5/2014 下午10:55
	ICT_Learner.zip	存档	15/12/2014 下午7:41

The file "ICT_Learner.zip" is circled in red at the bottom of the list.

The softcopies of "Testing and Evaluation Form" (shown below) is sent to the students, teachers.



A shared folder is made, for the testing of the condition of more than one user. I use it to test too. The emails of the students, teachers and programmer have given the authority to edit the files.



The softcopy of the Form and program is given to the programmer by a USB.

Testing and Evaluation Form

Please help to evaluate the program: Multiple-Choice Analysis. Thanks!

Instruction:

- Please execute the program by unzip the WHOLE folder (ICT_Learner) to the root of the hard disk(e.g. G:\)

Please download the program(*.zip) at https://www.dropbox.com/s/973ivva9sh24dv2/ICT_Learner.zip?dl=0

Report on Bugs:

No.	Description of errors

Program Evaluation:

Please answer the following questions by circling the numbers on the right hand side.

		Rating					Score
		Agree	Average	Disagree			
1.	The program is user-friendly	5	4	3	2	1	3.5
2.	The design of the input screen is good	5	4	3	2	1	4
3.	The format of the record is good	5	4	3	2	1	3
4.	The functions are useful for teaching/learning	5	4	3	2	1	5

Which features you like most?

What other functions should be provided by the program?

Other Suggestions

Evaluator's Name: _____ Date: _____



There are some positive comments:

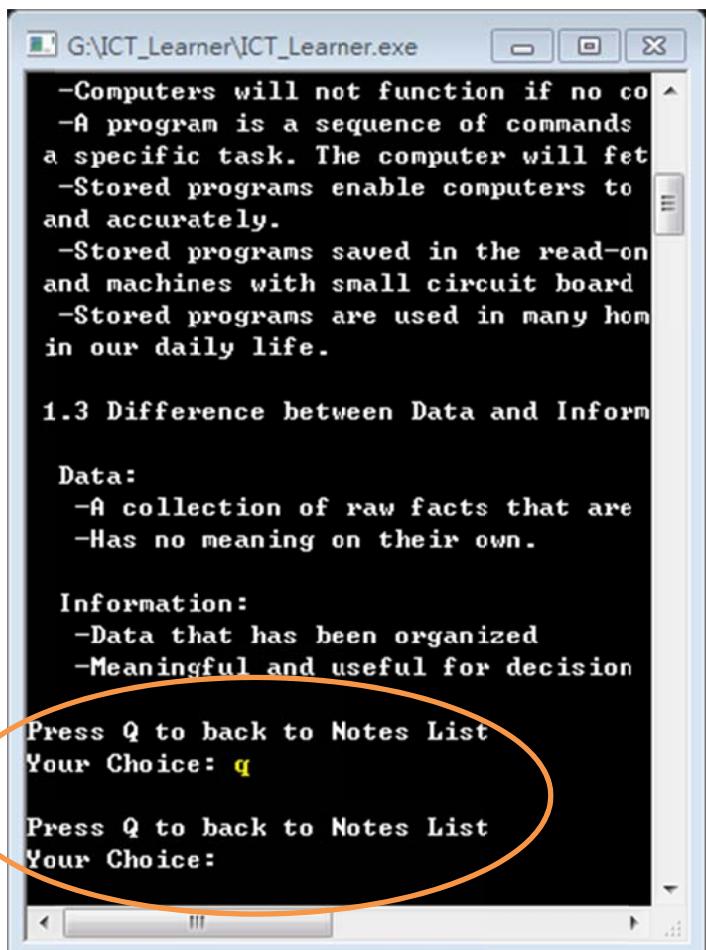
1. Most of the users like the game especially the matching game (17 out of 20)
2. All of the reviewers like the comment panel (20 out of 20)
3. "It was clear that the choice I chose is in color" (A student)
4. 'The style in the *.pas file is good, I can easily identify which parts are related' (The programmer)

However, some problems are found:

107

Problem 1 (The exit of the resources after usage)

The users reported that the 'q' are not accepted when they want to get back to the menu.



After modifying, the 'q' is also accepted in all the other panels when they finish using the resources.

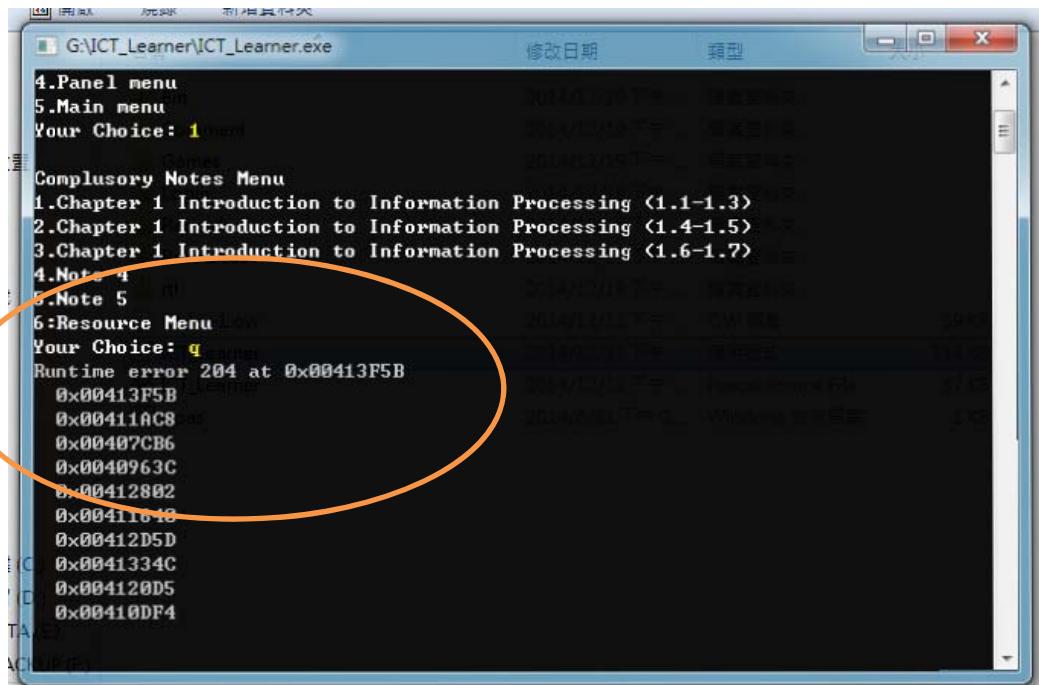
Problem 2 (The problem of mistyping in the choices procedures)

108

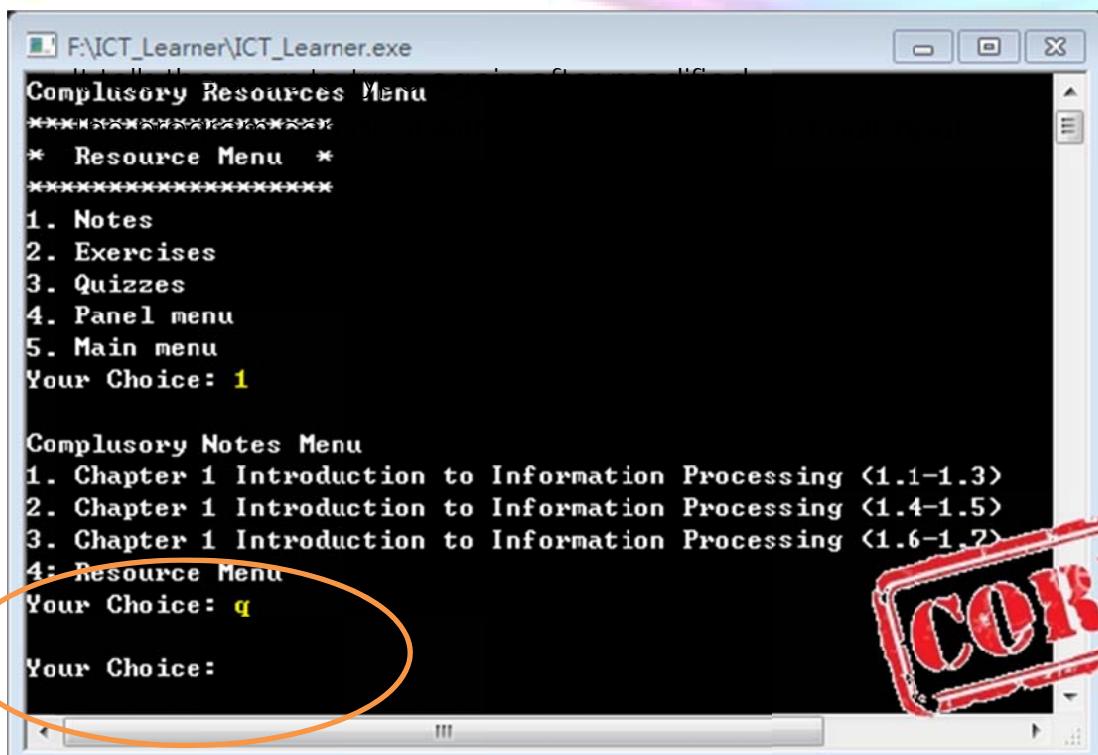
The reviewer found that there is a problem when typing in a letter or type nothing in a lot of procedures

It is an example from notes panel:

The program quit directly:



After modified:

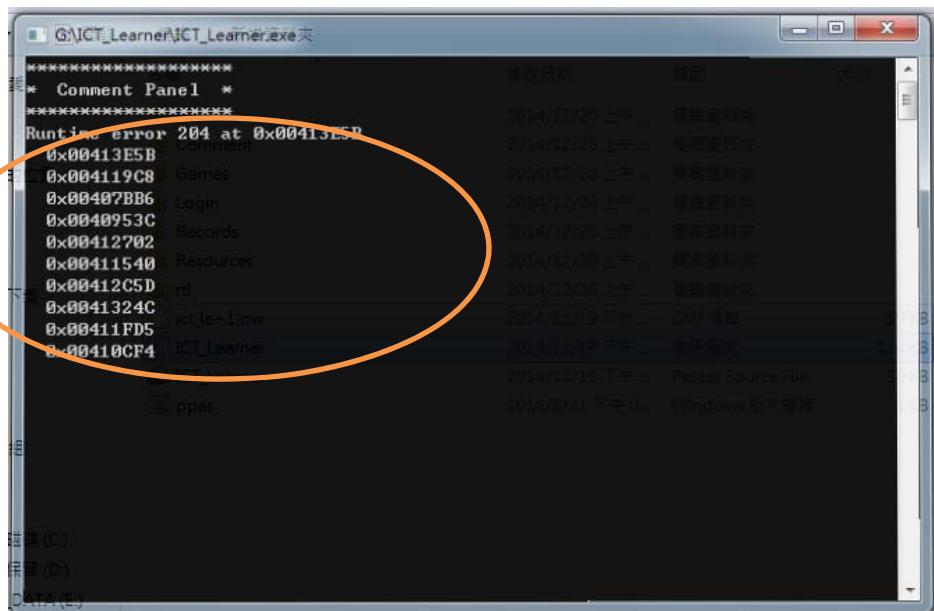


The choice procedures with same problem are also modified.

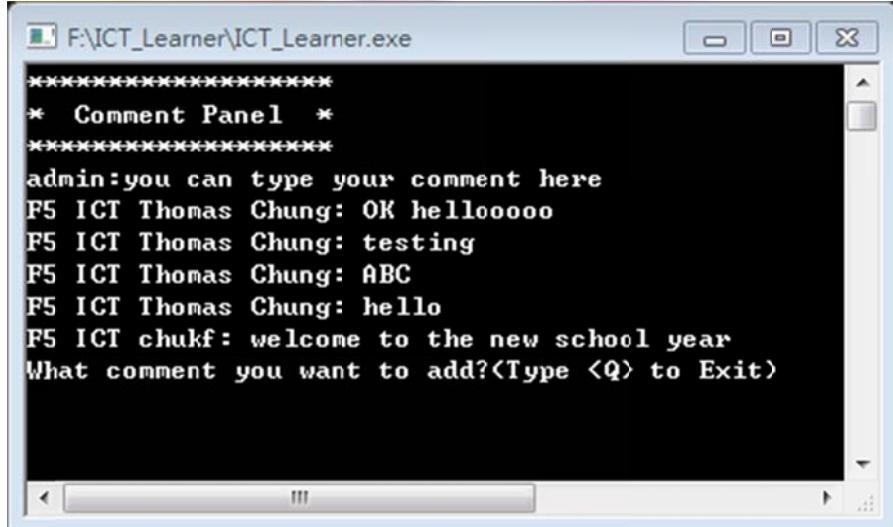
Problem 3 comment panel-the trouble of not accessible in the 2nd time.

The teacher reported that they cannot get in the comment panel twice in a single login time. ← 109

The screenshot of comment panel when get in 2nd time



After modified, the users can get into the comment panel a lot of time in one single login time.

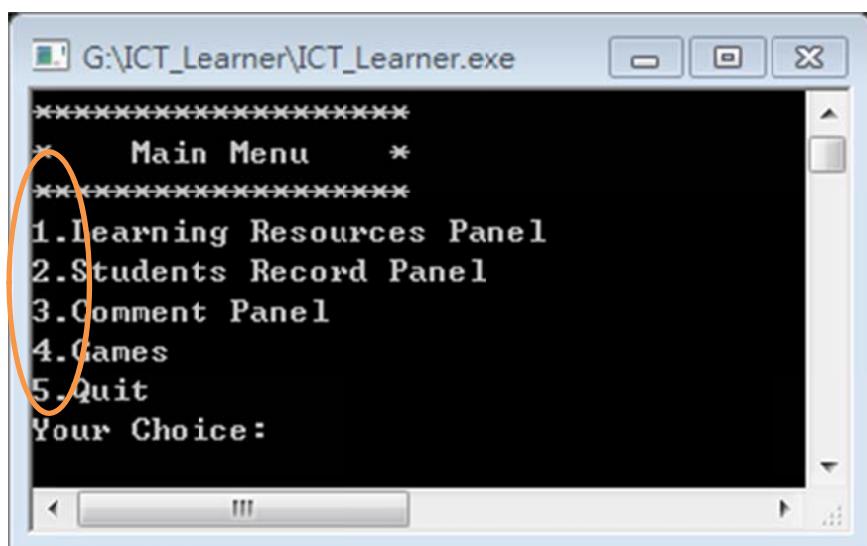


No more problems.

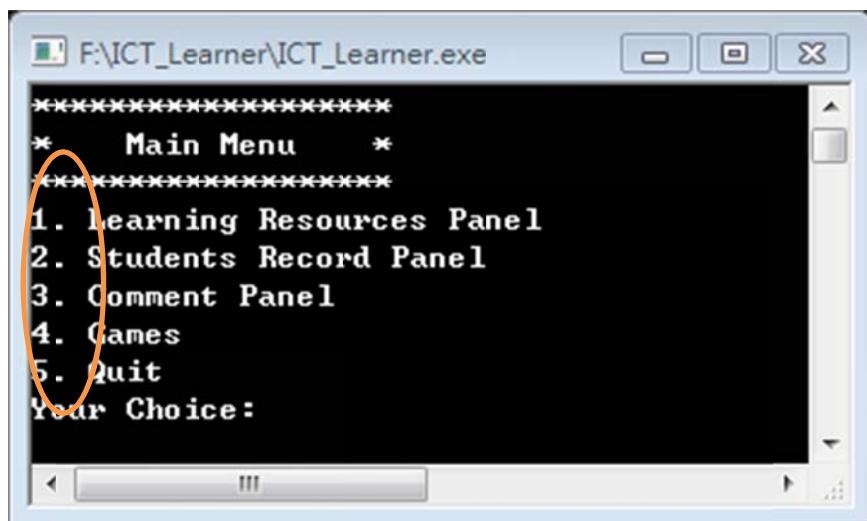
CORRECTED

Problem 4 spacement has been changed for almost all menus in the program

The teachers think that the spacement of the menu is not too good.



After modified:



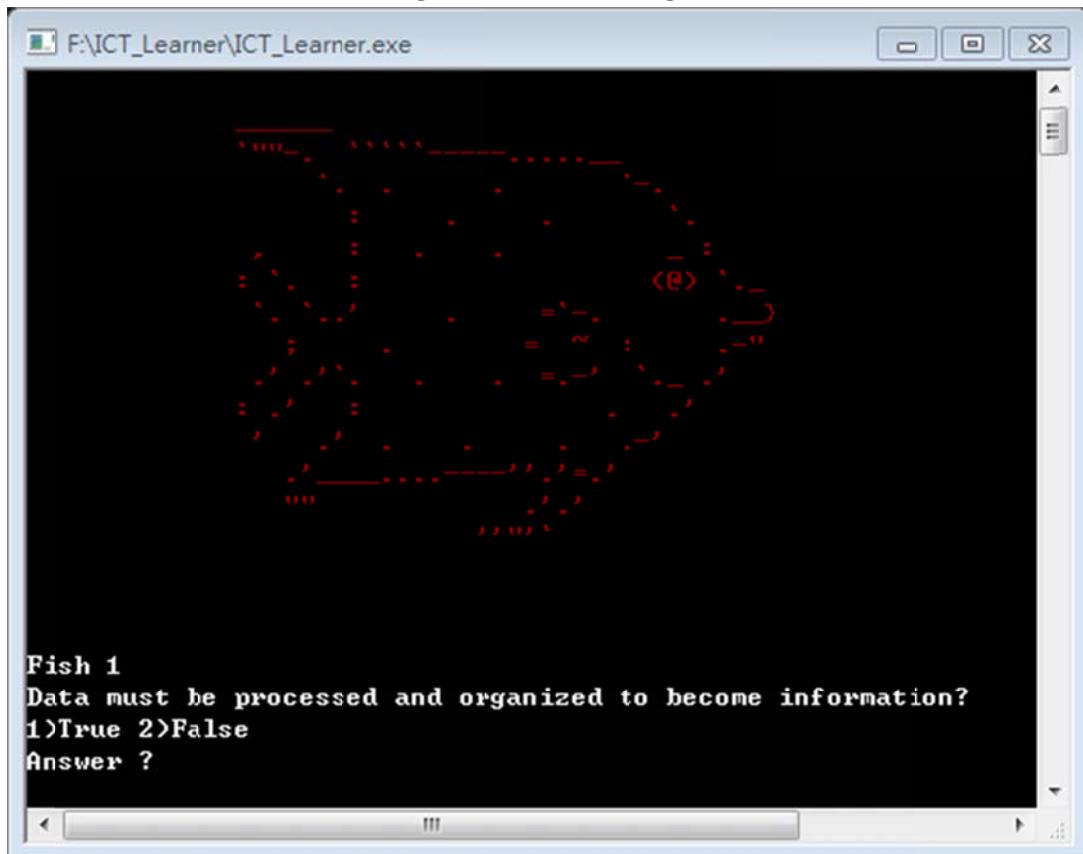
Almost all the menus are modified to have this better spacement.

It is more easy and clear to read the menu after modified.

CORRECTED

A teacher thinks that, the fishing game is well designed and fun. However, fish may not be too related to ICT study.

The program is made to suitable for the fishing game. Some followings may be done later, such as a fish with a gadget or change fishes to computer



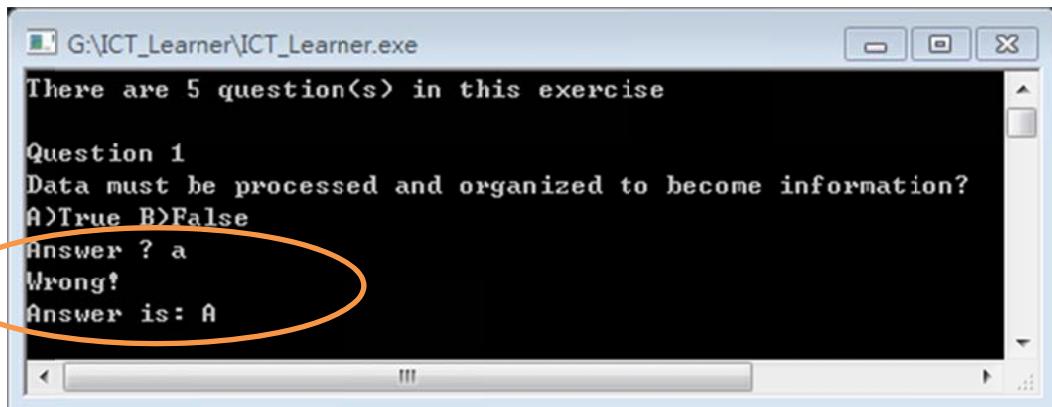
How the improvement be made will be considered, later. However, I think the fishing game is also good for not relating to ICT so much. Such that, the students can know that ICT is related to daily life and fun but not only studies and dull.



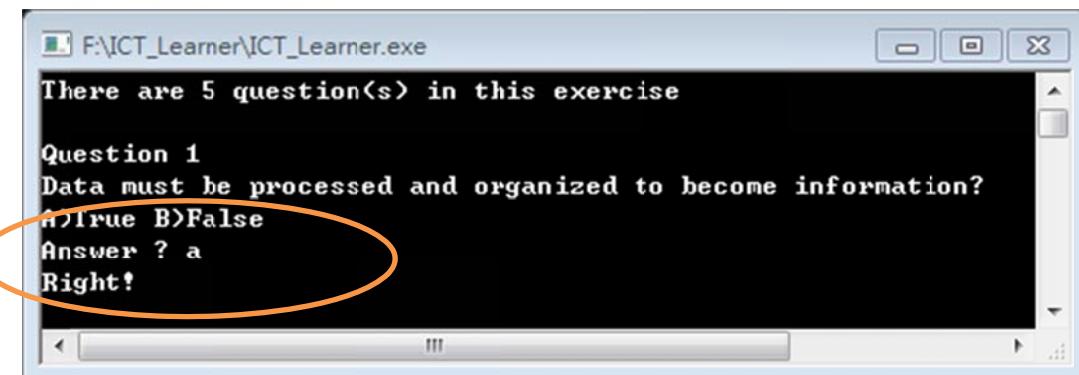
Problem 6 did not accept small letter in exercise and quiz

The students found that when they type small letters in exercises or quizzes. The answers are regarded as wrong. It may reduce their enthusiastic towards learning ICT. Therefore, the program is modified to accept them.

Before modify:



After modified:



After modified, both big and small letters are accepted. It won't bore the users that they are wrong if they type the small letter.

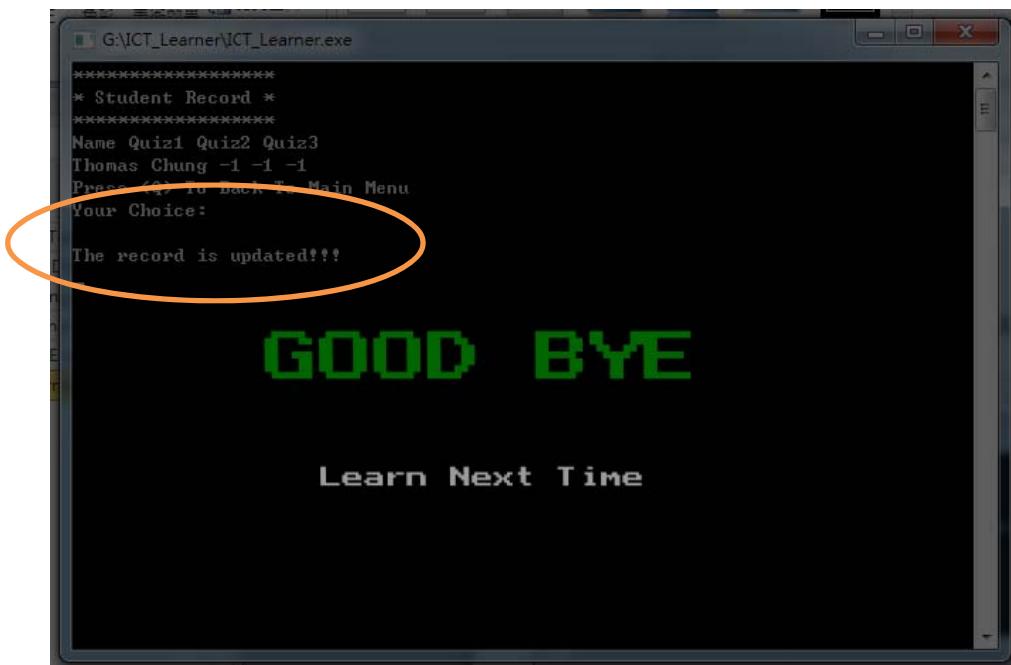
However, it may have a problem that if the blank in 'fill in the blanks question' is in the first position, the students may not care about the big letter and small letter in the real exams.



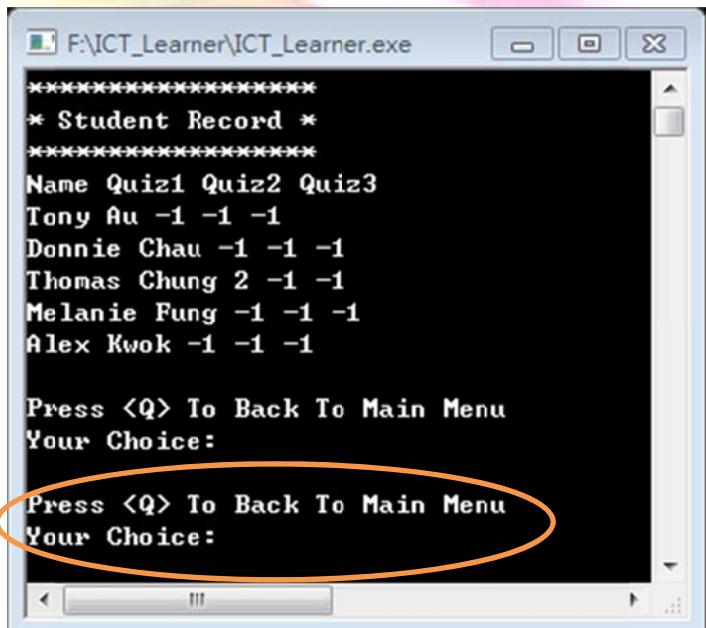
Problem 7 the student record panel exit the program directly when only 'Enter' is typed

113

The reviewer reported that when he types nothing in the student record page, the program quit.



After modified:



A statement is shown to tell the users to type again.

CORRECTED

5.5.1 Evaluation

In this program, some minor problems or interfaces are also fixed, the program has become better.

In the program, the condition of using the files together cannot be handled well. The records saving procedure have problems when they use it. The teachers should tell the students to log out one by one to solve the problems. It is hoped that the program can be modified to solve this problem automatically.

- Good features of the program:
 - ❖ The coding structure is good.
 - ❖ The procedures are good that is reusable
 - ❖ The interface is clear and simple
 - ❖ The games are worth playing.
 - ❖ The login procedure is good that the password can be seen
 - ❖ The menu showing is good.

- Shortcomings of the program:
 - ❖ The bugs when more than one user use the program
 - ❖ User-friendless
 - The design of the interface may be too dull.
 - Teacher play an important role in using this learner
 - ❖ Performance
 - Some parts are not too perfect
 - ❖ Flexibility for future development
 - There are limitation in the number of lines in the files
 - There are standard file format and structure for program.
 - ❖ Reusability of program codes
 - The procedures call through each other, it may not easily checked by the next programmer for the flow.

There can be still errors and potential bugs, an email will be set up later for the users to report the errors.

All in all, the program(CT_Learner) is finished at this moment.



Chapter 6 Conclusion & Discussion

6.1 Pros and cons of my Program

Pros and cons of my ICT_Learner

I have spent lots of time in doing this ICT_Learner. 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 users, which I cannot correct. There is showing my ICT_Learner's pros and cons.

My ICT_Learner is an educational system for senior students who study ICT. So, ICT's information and exercises are mainly recommended in this Learner. Students can learn more about ICT and discuss with other students and teachers. It is not an individual learning but they have interaction with others during their learning. They can enjoy the learning experiences and time with others rather than learning alone.

After using this learner, I think some of the students would improve their performance in ICT. The students can see if their understanding in ICT is right or not. The teachers can also see the performances of the students under a help of local public folder. Teachers can provide links which are useful and interesting. Students can save time to search for those web sites. I used a simple outline for the program such that students can easily find out the information they want.

In the program coding efficiency, almost all the files are read when they are needed, it may reduce time for reading the files in the start of the program. The lists are also read when they are needed, the teachers can edit the menu easily without the help from programmer and IT supporter as the menu is no need to change in the program. It is also a good point in users-friendliness.

My i-Learner can give students to have a better environment to learn ICT. Not just only focus on theories but also the practical and daily life.

Using this ICT_Learner, students can save time finding out the notes and they can read the notes when they are having holidays. The students can make use of the leisure time after school to know more ICT. The games in the learner also give a positive image to students that learning can be fun and easy. Therefore, the games are made to be refreshing and eye-catching.

It is hoped that the learning atmosphere among students can be amended.



Despite the benefits to students and teachers are great, there are still some inconveniences in the ICT_Learner. 116

In this learner, the support of teachers is important for the resources and the record checking system. It was discussed that the folder can be put in a public folder of a school server. There is a need for the authority of students and teachers to edit file. There is a problem of security. Teachers play a vital role in maintaining the whole system.

If the school has no internet access, there will be a problem. But in view of the widespread of gadgets, some of the schools are using notebook to teach. My learner can be used to support their teaching. The lowest requirement for better performance is 1024*768 resolution. I think users would not really mind to scroll down to read, but for some of cases, not all the words can be shown on the screen. And the program requires the users to put the folder in a root the hard disk, it may confuse the other users who are not using this program. However, it is more convenient for the users to put in the root of a hard disk, and the loading time would be faster too.

This is program which really has many tasks to read from files, if the hard disk reading speed is slow, it reduces its performance. Their loading time may be longer. I am really sorry about this.

For the quiz, record update function, the record is updated when the user log out. However, there will be error if more than 1 user log out at the same time. Therefore, teachers need to make special arrangement for the record update issue after finishing the quiz. It may be solved by, logging out start from num1...

There are also some copyright issues when copying the resources from books or internet. Therefore, it cannot be used for commercial use.

In the program coding efficiency, almost all the searching is searched by binary searching method, it may reduce the efficiency of the time for searching if there is a lot of information in the text file. There is problem that reading the files when it needed that, it may have problem during their usage, it may confuse them. While reading all file at first that avoids this problem. However, In view of the efficiency and the use of memory, I think using the initial method is better. Therefore, users (mainly teachers) should make sure the format and structure and the location of the folder is correct.

I hope I can do better next time, in order to make my ICT_Learner smoother and suitable for different kinds of people.

I hope my ICT_Learner can help the education process of the teachers and students. I hope I can help my next generation to know more about ICT. The youngers nowadays only focus on the games, I hope that the youngers can make use of the gadgets to know more about ICT and help to develop the society with their knowledge. I hope my ICT_Learner can be published and be used widely after having a better and new edition.



6.2 Future Improvement

In future, I really hope I can add more information into my ICT_Learner. So, students can know more about ICT all-roundly, not only on compulsory but also other electives. It will be good for my HKDSE, when I add more information in it, I can recall my memories of ICT.

As a student who is studying ICT, too. I may want to add some tips of learning ICT during the 3 past years. I hope I can add my experiences in the system to share it with my schoolmates. The section may be called "The words from big brothers"

And there is a account for admin, however, the functions of admin is not enough because there are no classes allocated for the admin. I hope I can improve the powerfulness of the Admin account but not just a dummy to test small part but also all. The admin account is an important role to test out the program.

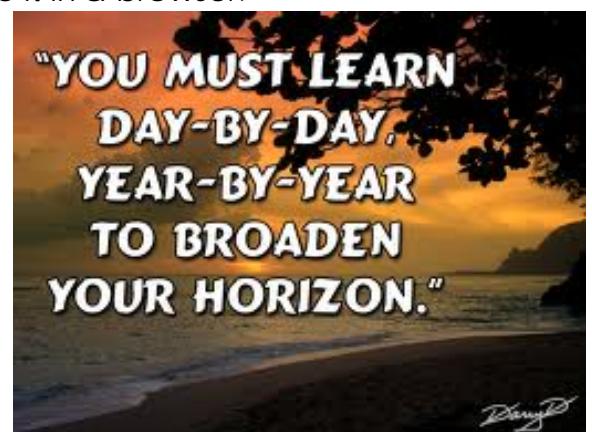
The editing functions of the teachers are not very well-developed. It depends on the teachers to edit by text with the help of user guide. I hope to add functions to the program such that the teachers can edit the resources in the system but not externally. In my project, I didn't do so, as I think different schools have different requirement for teachers. Therefore, I make it free for the next programmers to create their own functions with creativity.

Also, I will add in more games so that the ICT_Learner can have a funnier impression. I will also edit the fishing game to make it more related to ICT. The games are expected to be designed to relate to ICT more. And some games just for entertainment will be developed for students as well as teachers to relax from work. It is hoped that the students and teachers won't bury heads in but also spend time for entertainment to balance between work and games.

If there is a support of large database, I hope to develop a timetable functions for the students to use. Not only does it useful for their studies in ICT, but it also help the performance of students in other subjects.

Also, having launch the online servers of schools, schools can simply make a webpage and add the program in it. However, it needs the support of webpage programmer to link the internet with the ICT_Learner. With this support, the students can use the ICT_Learner anywhere, anytime and any devices as they can use it in a browser.

I would try to add in different electives. For example, I will find out some information about the Internet, Website developing etc. So that different and more users can use this not only in my school but also the other schools. In the process, I may also learn about the thing about the other elective not just about programming. It can broaden my horizons during the process.



In order to shorten the searching time, I would try more different kinds of method such as binary searching, merge searching, etc. The program is going to be handled over to the other programmer for further development and improvement for their own schools and specifies the functions for their own schools own purposes. The code will be given to the IT supporter of the school in request. In this project, the *.pas is also in the folder of "ICT_Learner".

Before handing over the program to the next programmer of IT supporter, I may make a programmer guideline for them. It will be done when they request for it.

As, the comments in the chapter 5 stated, the programmer think it is easy to chase different part of the functions and the codes style are well defined. There are different short descriptions for the programmer to understand what are the procedures writing for. Therefore, it is easy for the further development by the next programmer.



Not only students, but also other aged people would like to know about ICT in this IT Age. So, I hope to provide some other ICT_Learner versions for adults or even children. Give them a chance to know about ICT. I hope there will be more editions for different cities. It may be adding procedure in the system for users to choose their language. I hope the use of ICT_Learner would be worldwide.

With the use of ICT_Learner, it is hoped that the knowledge towards ICT of citizens can be enhanced.



6.3 Self-Reflection

In making this ICT_Learner, I have learnt more about computer, software development. Also, I gain some life-long knowledge, skills and morals.

I have understood the mystery of ICT. In the previous sections, I found that there are still a lot in ICT. What we learn in the secondary school is just part of it. In this state in art era, the technology of ICT is improving. We always think that we know about the world and control the improvement of technology. Like what Prof Stephen Hawking said in a BBC program, he fears artificial intelligence (AI) could spell the end of humanity. The AI can improve them and the human cannot even compete with them at last. In fact, nowadays, a myriad of people are controlled by their phones and the games.

Will the day really come true? I think there is a probability. But it is avoidable, we should understand the mystery of ICT and face the problem. The knowledge has no end, the more we know, the more we don't know. In developing the system, I learnt about the knowledge of ICT that cannot be learnt in the classroom. We should always run out of the classroom and find new knowledge.

I have learnt more about how to develop software, collect useful data, ways to get information. These are useful for me.

I have known that developing software is not just typing software. It involves a lot of designs, analyzing, doing project .etc. A well software is not only valued with this functions but also its report, the theories behind. Software development is not like the common tasks in classes, it is a large development which needs a lot of time from the beginning to the end. While developing software, lots of things need to be considered, the user requirement and the interface. I learnt that the coding style is very important that it is used for the checking and further development.

I also found the value of the print and books. We usually search things online nowadays, however, the black in white is not invaluable. They are worth-reading. From the book "print is dead" by Jeff Gomez. He said that there is a still place for the paper and books. I cannot agree more with it, in my ICT_learner, it turns books into virtual and can be read on devices. However, real life printing is also need to support it such as printing out the accounts and passwords for the students and the records of students. And when they having class, they use pen and paper to dot notes down .Print is not dead, nevertheless, it turns to another form of new technology with the e-books.

There are some useful ways to collect data and information such as interviewing with others, reviewing others' programs or reading books. I found that the data need to be analyzed before being information. Comparing different information is very important to know the things behind. Getting information is not easy as I thought before. We should always read the information carefully and find out the useful content. Try to understand the words and write in our own words. Copying is just the data, what we write is the information

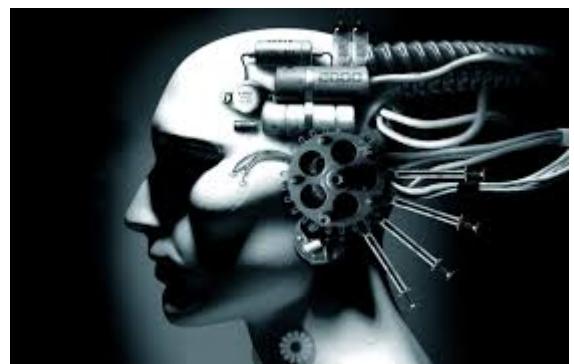
Not only did I learn about the academic things and skills, I have also learnt about some life-long skills and morals.

I have understood that we should be enthusiastic when doing things. In this project, software development can be dull to people that don't like programming. But for me, I treat it as an entertainment and learning process. I enjoy the process of writing the system. It is hard for us to keep up doing things if we don't like it. We should love what we are doing. Try not to do what we like, sometimes we have to love what we need to do. It helps in my life that I need to be happy when doing things. If we don't enjoy what we are doing, we will feel unhappy and cannot do the job well.

For all long software, it needs a lot of time to develop. Some of the systems like Windows are developed by a team. In this project, I worked alone, however, I can seek help from friends and teachers. We should bear in mind that we are not alone. And the most important things that I have learnt is to have continuous improvement. Like what I mentioned, people do not realize the power of the computer and its potential that we can always control it. What is in fact that we can't even control ourselves.

We cannot argue that we may better than AI. As, we always think we are the best and not to improve ourselves. However, AI did. But when we think about the humanity, AI has no humanity. What we have to do is to improve ourselves continuously. If the AI is developed, will we have the power to control them? If we still rest complacently on our laurels, the AI will able to destroy our humanity. Like my ICT_learner, without the self-control of the users, it is just entertainment software but if people make use of it, it is a good educational system!

We need to learn more, read more, and know more about the world. We can be powerful, if not strongest type of AI. With the high level technology and our humanity, WE, human, indeed can become the best AI



I hope to express my great thanks to my teachers for giving a chance to do this project, it is an fruitful, valuable and worthy project----- ICT_Learner-----An educational system.

Chapter 7 Reference and Acknowledgement#

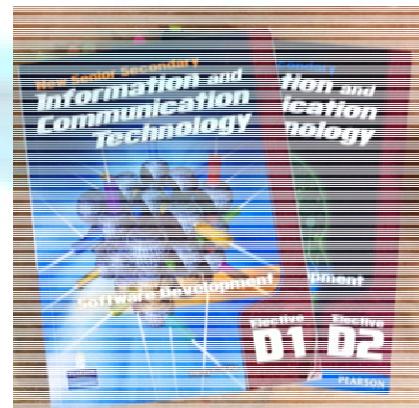
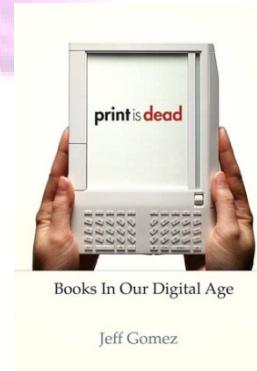
From Internet websites

1. <http://reocities.com/SoHo/7373/aquatic.htm>
2. <https://www.google.com.hk>
3. <http://programmersheaven.com/categories/pasprog>
4. <http://computer-programming-forum.com/29-pascal/b983fb10db5ff8ea.htm>
5. <http://www.freepascal.org/docs-html/rtl/graph/index.html>
6. <http://pascal-programming.info/lessonindex.html>
7. <http://www.bbc.com/news/technology-30293863>



From books

1. 'print is dead' by Jeff Gomez
2. NSS Information and Communication Technology D1
3. NSS Information and Communication Technology D2
4. Some report samples.
5. 201 principles of software development / Alan M. Michael.



Acknowledgement

1. Thanks Mr. Chu for his help
2. Thanks my students for helping evaluate the program
3. Thanks the programmer reviewer Mr. Chung for helping me out in the evaluation



Appendices

Appendix 1 – Program Code (after Testing & Evaluation)

```

{*****
* This Program Is Designed and Written By:
* Cheung Sha Wan Catholic Secondary School
* Chung Ho Yin 6A 7 2014-2015
* Supervising Teacher:Mr. Chu
* Final Edition Date:20-12-2014
*****}

program ICT_Learner;
uses crt,wincrt,Graph;
type
recordtype = record
    Name: string;
    quiz1 ,quiz2 , quiz3 :integer;
end;
var
    choice,choiceL,valid_account,s_num,i,j:integer;
    login_choice,HDD,choiceC,c_name,Student_Class,Teacher_Class,status:string;
    choice_chr:char;
    {Graphic}
    GraphicsDriver, GraphicsMode: smallint;
    {Student Record}
    Student:array[1..30] of recordtype;
    {Matching Game Variables}
    num_of_card,correct_match,num_of_pairs,fliped_times:integer;
    card_content:array[1..9,1..8] of string;
    answer_match:array[1..4]of string;
    {Car Racing Variables}
    CN,CK,CNC,CL:integer;
    CQ,CCH,CANS:array[1..5] of string;

{*****
* Assign all the procedure and function for later usage *
*****}

procedure Welcome_Page;forward;
procedure Good_Bye_Page;forward;
procedure GetDriveName;forward;
procedure Make_choice;forward;

```

```
procedure Make_choice_chr;forward;
{Login}
procedure Login; forward;
procedure Login_T; forward;
procedure Login_S; forward;
procedure Students_Choices_List; forward;
procedure Teachers_Choices_List; forward;
{Menus}
procedure Main_Menu;forward;
procedure Resources_Menu;forward;
procedure Games_Menu;forward;
{Resources}
procedure Complusory_OR_Elective; forward;
procedure Complusory; forward;
procedure Elective; forward;
procedure Resources_List(list_file:string);forward;
procedure Resources_List_C; forward;
procedure Resources_List_E; forward;
procedure Complusory_Notes_List; forward;
procedure Complusory_Exercises_List; forward;
procedure Complusory_Quizzes_List; forward;
procedure Elective_Notes_List; forward;
procedure Elective_Exercises_List; forward;
procedure Elective_Quizzes_List; forward;
{Use resources}
procedure Read_Note(notenum:string);forward;
procedure Do_Ex(ex_num:string);forward;
procedure Do_Quiz(quiznum:string);forward;
{Record}
procedure Store_Record;forward;
procedure Show_Record; forward;
procedure Update_Record(quiz_done:char; score_record:integer); forward;
procedure Update_Record_File;forward;
{Comments}
procedure Comment_Panel;forward;forward;
{Games}
procedure VocabGuessGame;forward;
procedure Matching_Game;forward;
procedure Fishing_Game;forward;
procedure Car_Racing_Game;forward;
```

```

{*****-----Procedures-----*}
* Procedures
*****-----} }

{****-----Welcome Page-----***}
procedure Welcome_Page;
var graphkey:char;
begin
GraphicsDriver := Detect;
InitGraph(GraphicsDriver, GraphicsMode, '');
SetColor(Green);
settextstyle(TriplexFont,HorizDir,5);
Outtextxy(70,300,'Welcome To ICT Learner');
Delay(3000);
SetColor(White);
SetTextStyle(defaultfont, HorizDir,2);
Outtextxy(270,400,'Type Any Key To Start Learning');
graphkey:=readkey;
if graphkey<>#0 then
  closegraph;
end;
{****-----End of Welcome Page-----***}

{****-----Good Bye Page-----***}
procedure Good_Bye_Page;
begin
GraphicsDriver := Detect;
InitGraph(GraphicsDriver, GraphicsMode, '');
SetColor(Green);
settextstyle(TriplexFont,HorizDir,5);
Outtextxy(230,300,'GOOD BYE');
SetColor(White);
SetTextStyle(defaultfont, HorizDir,2);
Outtextxy(270,400,'Learn Next Time');
delay(1000);
closegraph;
end;
{****-----End of Good Bye Page-----***}

```

```
{*****---Get Drive Name---****}
procedure GetDriveName;
begin
HDD:=(Copy((ParamStr(0)), 1, 2)+ '\');
writeln('The Directory Is Located In ', HDD);
end;
{*****---End Of Getting Drive Name---****}
```

{*****---Choice Making procedure---****}

```
procedure Make_choice;
begin
write('Your Choice: ');
textcolor(14);
readln(choice);
textcolor(15);
writeln
end;
```

```
procedure Make_choice_chr;
begin
write('Your Choice: ');
textcolor(14);
readln(choice_chr);
textcolor(15);
writeln
end;
{*****---End of Choice Making procedure---****}
```

```
{*****
*      Login process      *
*****}
```

```
procedure Login;
begin
repeat
  Clrscr;
  valid_account := 0;
  writeln('Type S for students, Type T For teachers');
  write('Your Choice: ');
  textcolor(14);
  readln(login_choice);
  textcolor(15);
```

```

until (login_choice = 's') OR (login_choice = 't')OR (login_choice = 'T')
OR (login_choice = 'S')OR (login_choice = 'admin');
login_choice := upcase(login_choice);
{admin is for the programmer to have a quick access to program to test}
if login_choice = 'S' then
  Login_S;
if login_choice = 'T' then
  Login_T;
if login_choice = 'ADMIN' then
  Teachers_Choices_List;
if valid_account = 1 then
begin
  if login_choice = 'S' then
    Students_Choices_List;
  if login_choice = 'T' then
    Teachers_Choices_List;
end;
end;
{-----}
procedure Login_T;
var form,name,filename,pw,account,quit,ac:string;
  login_file:text;
begin
repeat
  Clrscr;
  writeln('*****');
  Writeln('* Teacher Login *');
  writeln('*****');
  form:='';
  name:='';
  pw:='';
  account:='';
  filename:='';
  c_name:=' ';
  valid_account := 0;
repeat
  Write('What is your form?(e.g.4):');
  textColor(14);
  readln(form);
  textColor(15);
until (form >='4') and (form <= '6');
Write('What is your name?:');

```

```
textcolor(14);
readln(name);
textcolor(15);
c_name:= name;
Write('Password:');
textcolor(14);
readln(pw);
textcolor(15);
filename:='F'+ form +'.txt';
account:= name+ ' ' + pw;
assign(login_file,HDD+'ICT_Learner\Login\Teacher_Accounts\' + filename);
reset(login_file);
while not eof(login_file) do
begin
  readln(login_file,ac);
  if account = ac then
    begin
      valid_account := 1;
      Teacher_Class:='F'+ form;
      status:='T';
      Store_Record;
    end;
  end;
  if valid_account = 0 then
  begin
    textcolor(4);
    writeln('Invalid Account!');
    write('Quit?(Y/N):');
    readln(quit);
    textcolor(15);
  end;
close(login_file);
until (valid_account = 1) OR (quit = 'Y');
end;
{-----}
procedure Login_S;
var form,name,filename,pw,account,quit,ac:string;
  login_file:text;
begin
repeat
  Clrscr;
  writeln('*****');

```

```
Writeln('* Student Login *');
writeln('*****');
form:='';
name:='';
pw:='';
account:='';
filename:='';
valid_account := 0;
repeat
  Write('What is your form?(e.g.4):');
  textColor(14);
  readln(form);
  textColor(15);
until (form >= '4') and (form <= '6');
Write('What is your name?:');
textcolor(14);
readln(name);
textcolor(15);
c_name:= name;
Write('Password:');
textcolor(14);
readln(pw);
textcolor(15);
filename:='F'+ form +' ICT.txt';
account:= name+ ' ' + pw;
assign(login_file,HDD+'ICT_Learner\Login\Student_Accounts\' + filename);
reset(login_file);
while not eof(login_file) do
begin
  readln(login_file,ac);
  if account = ac then
  begin
    valid_account := 1;
    Student_Class:='F'+ form +' ICT';
    status:='S';
    Store_Record;
  end;
end;
if valid_account = 0 then
begin
  textColor(4);
  writeln('Invalid Account!');
end;
```

```

        write('Quit?(Y/N):');
        readln(quit);
        textcolor(15);
        end;
        close(login_file);
until (valid_account = 1) OR (quit = 'Y');

end;
{-----End Of Login Procedure-----}

{*****}
* Choice Panel *
{*****}

{-NO such panel for the users, just the functions use for choice-}
{-Just to let the next programmer know-}

{---Show Lists of choices in Main Menu---}

procedure Students_Choices_List;
begin
repeat
  Clrscr;
  Main_Menu;
  choice:=0;
  Make_choice_chr;
until (ord(choice_chr)>= 49) and (ord(choice_chr) <= 53);
choice:= ord(choice_chr)-48;
case choice of
  1:Complusory_OR_Elective;
  2>Show_Record;
  3>Comment_Panel;
  4>Games_Menu;
end
end;
{-----}

procedure Teachers_Choices_List;
begin
repeat
  Clrscr;
  Main_Menu;
  choice:=0;
  Make_choice_chr;

```

```
until (ord(choice_chr)>= 49) and (ord(choice_chr) <= 53);
choice:= ord(choice_chr)-48;
case choice of
  1:Complusory_OR_Elective;
  2>Show_Record;
  3:Comment_Panel;
  4:Games_Menu;
end
end;

{*****-----Menus-----*****}

procedure Main_Menu;
begin
writeln('*****');
writeln('* Main Menu *');
writeln('*****');
Writeln('1. Learning Resources Panel');
Writeln('2. Students Record Panel');
Writeln('3. Comment Panel ');
Writeln('4. Games');
Writeln('5. Quit');
end;
{-----}

procedure Resources_Menu;
begin
writeln('*****');
writeln('* Resource Menu *');
writeln('*****');
if status='T' then
begin
writeln('10. Edit Notes');
writeln('20. Edit Exercises');
writeln('30. Edit Quizzes');
end;
writeln('1. Notes');
writeln('2. Exercises');
writeln('3. Quizzes');
writeln('4. Panel menu');
writeln('5. Main menu');
end;
{-----}

procedure Games_Menu;
begin
```

```

repeat
choice :=0;
clrscr;
writeln('*****');
writeln('*      Game Menu      *');
writeln('*****');
writeln('1. Vocabulary Guessing Game');
writeln('2. Matching Game');
writeln('3. Fishing Game');
writeln('4. Car Racing Game');
writeln('5. Main Menu');
Make_choice;
until ((choice >= 1) and (choice <= 5));
case choice of
1:VocabGuessGame;
2:Matching_Game;
3:Fishing_Game;
4:Car_Racing_Game;
5:begin
if login_choice = 'S' then
  Students_Choices_List;
if login_choice = 'T' then
  Teachers_Choices_List;
end;
end;
end;

{*****
*      Resource Panel      *
*****}

{-----Choose Complusory or elective-----}
procedure Complusory_OR_Elective;
begin
repeat
Clrscr;
writeln('Learning Resources Panel Menu');
writeln('1. Complusory');
writeln('2. Elective');
Make_choice
Until (choice = 1) or (choice = 2);
case choice of

```

```

1:Complusory;
2:Elective

end;
end;

{-----}
procedure Complusory;
begin
  Resources_List_C;
end;

{----- # Elective part can be further developed -----}

procedure Elective;
begin
  {Resources_List_E;}
  Writeln('Go To The Following Link For Notes');
  writeln('http://pascal-programming.info/index.php');
  writeln;
  repeat
    writeln('Press <Q> To Back To Main Menu');
    Make_choice_chr;
    until choice_chr in['Q','q'];
    if choice_chr in['Q','q'] then
      begin
        if login_choice = 'S' then
          Students_Choices_List
        else
          Teachers_Choices_List
      end;
  end;

{-----Resources_Choice-----}

procedure Resources_List(list_file:string);
var count:integer;
  content:string;
  Resources_List:text;
begin
  assign(Resources_List,HDD+'ICT_Learner\Resources\' +list_file+'.txt');
  reset(Resources_List);
  count:=0;
  while not eof(Resources_List) do
    begin
      readln(Resources_List,content);
      writeln(content);
    end;
end;

```

```

count:= count + 1
end;
writeln(count,': Resource Menu');
choiceL := count;
close(Resources_List);
end;
{-----
-Show Resources Lists-
-----}

procedure Resources_List_C;
begin
Clrscr;
writeln('Complusory Resources Menu');
Resources_Menu;
choice:=0;
repeat
Make_choice;
until ((choice >= 1)and (choice <= 5))or (choice =10)or (choice =20)or (choice=30);
case choice of
1:Complusory_Notes_List;
2:Complusory_Exercises_List;
3:Complusory_Quizzes_List;
4:Complusory_OR_Elective;
5:if login_choice = 'S' then
    Students_Choices_List
else
    Teachers_Choices_List;
10:writeln('Go to ',HDD+'ICT_Learner\Resources\Notes\Complusory',' To Edit');
20:writeln('Go to ',HDD+'ICT_Learner\Resources\Exercises\Complusory',' To Edit');
30:writeln('Go to ',HDD+'ICT_Learner\Resources\Quizzes\Complusory',' To Edit');
end;
if (choice = 10) or (choice = 20) or (choice = 30) then
begin
repeat
writeln('Press Q to back to Main_Menu');
Make_choice_chr;
until choice_chr in['Q','q'];
if choice_chr in['Q','q'] then
if login_choice = 'S' then
    Students_Choices_List;
if login_choice = 'T' then

```

```

Teachers_Choices_List;
end;
end;
{-----}
procedure Resources_List_E;
begin
Clrscr;
writeln('Elective Resources Menu');
Resources_Menu;
Make_choice;
case choice of
 1:Elective_Notes_List;
 2:Elective_Exercises_List;
 3:Elective_Quizzes_List;
 4:Complusory_OR_Elective;
 5:if login_choice = 'S' then
    Students_Choices_List
  else
    Teachers_Choices_List;
10:writeln('Go to',HDD+'ICT_Learner\Resources\Notes\Elective','To Edit');
20:writeln('Go to',HDD+'ICT_Learner\Resources\Exercises\Elective','To Edit');
30:writeln('Go to',HDD+'ICT_Learner\Resources\Quizzes\Elective','To Edit');
end;
end;
{-----}

procedure Complusory_Notes_List;
var notechoice:string;
begin
Resources_List('Note_C_List');
choice:=0;
repeat
  Make_choice_chr;
  until (choice_chr >= '1') and (choice_chr <= chr(choiceL+48)) ;
  choiceC:='Complusory';
  notechoice := choice_chr;
  if choice_chr = chr(choiceL+48) then
    Resources_List_C
  else
    Read_Note(notechoice);
end;

{-----}

```

```

procedure Complusory_Exercises_List;
var exchoice:string;
begin
  Resources_List('Ex_C_List');
  repeat
    Make_choice_chr;
    until (choice_chr >= '1') and (choice_chr <= chr(choiceL+48)) ;
    choiceC:='Complusory';
    exchoice := choice_chr;
    if choice_chr = chr(choiceL+48) then
      Resources_List_C
    else
      Do_Ex(exchoice);
  end;

{-----}

procedure Complusory_Quizzes_List;
var quizchoice:string;
begin
  Resources_List('Quiz_C_List');
  repeat
    Make_choice_chr;
    until (choice_chr >= '1') and (choice_chr <= chr(choiceL+48)) ;
    choiceC:='Complusory';
    quizchoice := choice_chr;
    if choice_chr = chr(choiceL+48) then
      Resources_List_C
    else
      Do_Quiz(quizchoice);
  end;

{-----# Elective is ready for further developed #-----}
procedure Elective_Notes_List;
var notechoice:string;
begin
  Resources_List('Note_E_List');
  choice:=0;
  repeat
    Make_choice_chr;
    until (choice_chr >= '1') and (choice_chr <= chr(choiceL+48)) ;
    choiceC:='Elective';
    notechoice := choice_chr;

```

```

if choice_chr = chr(choiceL+48) then
    Resources_List_E
else
    Read_Note(notechoice);
end;

{-----}
procedure Elective_Exercises_List;
var exchoice:string;
begin
    Resources_List('Ex_E_List');
    choice:=0;
repeat
    Make_choice_chr;
until (choice_chr >= '1') and (choice_chr <= chr(choiceL+48)) ;
choiceC:='Elective';
exchoice := choice_chr;
if choice_chr = chr(choiceL+48) then
    Resources_List_E
else
    Do_Ex(exchoice);
end;

{-----}
procedure Elective_Quizzes_List;
{var quizchoice:string;}
begin
    Resources_List('Quiz_E_List');
    choice:=0;
repeat
    Make_choice_chr;
until (choice_chr >= '1') and (choice_chr <= chr(choiceL+48)) ;
choiceC:='Elective';
if choice_chr = chr(choiceL+48) then
    Resources_List_E;
end;

{-----}
-   Use Resources   -
-----}

{----Read_Notes----}

```

```

procedure Read_Note(notenum:string);
var note_content:string;
    note_file:text;
begin
    Clrscr;
    assign(note_file,HDD+'\ICT_Learner\Resources\Notes\' + choiceC + '\Note' + notenum+'.txt');
    reset(note_file);
    while not eof(note_file) do
        begin
            readln(note_file,note_content);
            writeln(note_content);
        end;
    close(note_file);
    writeln;
repeat
    writeln('Press Q to back to Notes List');
    Make_choice_chr;
until choice_chr in ['Q','q'];
if choice_chr in ['Q','q'] then
begin
    Clrscr;
    if choiceC = 'Complusory' then
        Complusory_Notes_List
    else Elective_Notes_List;
end
end;

{----Do_Exercises----}
procedure Do_Ex(ex_num:string);
var
    N,K,NC,L:integer;
    QA:string;
    ex_file:text;
    QU,CH,AN :array[1..5] of string;
begin
    Clrscr;
    assign(ex_file,HDD+'\ICT_Learner\Resources\Exercises\' + choiceC + '\Ex' + ex_num+'.txt');
    reset(ex_file);
    readln(ex_file,N);
    for L := 1 to N do
        begin
            readln(ex_file,QU[L]);
        end;

```

```

readln(ex_file,CH[L]);
readln(ex_file,AN[L])
end;
close(ex_file);
{---Start Ex---}
writeln('There are ', N, ' question(s) in this exercise');
writeln;
NC:=0;
for K:= 1 to N do
begin
  writeln('Question ',K);
  writeln(QU[K]);
  writeln(CH[K]);
  repeat
    write('Answer ? ');
    readln(QA);
    until QA<> '';
  if upcase(QA) = upcase(AN[K]) then
  begin
    writeln('Right!');
    writeln;
    NC := NC + 1
  end
  else
  begin
    writeln('Wrong!');
    writeln('Answer is: ',AN[K]);
    writeln;
  end
end;
writeln('You have ', NC, ' mark(s)');
writeln;
repeat
writeln('Press Q to back to Exercises List');
Make_choice_chr;
until choice_chr in ['Q','q'];
if choice_chr in ['Q','q'] then
begin
  Clrscr;
  if choiceC = 'Complusory' then
    Complusory_Exercises_List
  else Elective_Exercises_List;

```

```

end;
end;

{-----Do_Quizzes----}
procedure Do_Quiz(quiznum:string);
var quiz_file:text;
QN,QK,QNC,QL:integer;
QQA:string;
QUES,CHO,ANS :array[1..5] of string;
begin
  Clrscr;
  assign(quiz_file,HDD+' ICT_Learner\Resources\Quizzes\' + choiceC+'\Quiz'+ quiznum+'.txt');
  reset(quiz_file);
  readln(quiz_file,QN);
  for QL := 1 to QN do
    begin
      readln(quiz_file,QUES[QL]);
      readln(quiz_file,CHO[QL]);
      readln(quiz_file,ANS[QL])
    end;
  close(quiz_file);
{---Start Quiz---}
writeln('There are ', QN, ' questions in this quiz');
writeln;
QNC:=0;
for QK:= 1 to QN do
begin
  writeln('Question ',QK);
  writeln(QUES[QK]);
  writeln(CHO[QK]);
  repeat
    write('Answer ? ');
    readln(QQA);
  until QQA<> '';
  if upcase(QQA) = upcase(ANS[QK])then
    begin
      writeln('Right!');
      writeln;
      QNC := QNC + 1
    end
    else
    begin
      writeln('Wrong!');
    end
  end
end;

```

```

writeln;
end
end;
writeln('You have ', QNC, ' mark(s)');
Update_Record(quiznum[1], QNC);
writeln('Answer will be given by teachers');
writeln;
repeat
writeln('Press Q to back to Quizzes List');
Make_choice_chr;
until choice_chr in ['Q','q'] ;
if choice_chr in ['Q','q'] then
begin
Clrscr;
if choiceC = 'Complusory' then
    Complusory_Quizzes_List
else
    Elective_Quizzes_List;
end;
end;
{*****
* Record Panel *
*****}

procedure Store_Record;
var record_file:text;
record_class:string;
begin
if status<>'T' then
    record_class := Student_Class
else
    record_class := Teacher_Class + ' ICT';

assign(record_file,HDD+'\ICT_Learner\Records\' + record_class +'.txt');
reset(record_file);
readln(record_file);
readln(record_file);
i:=0;
while not eof(record_file) do
begin
    i:=i+1;

```

```
with Student[i] do
begin
  readln(record_file, Name);
  readln(record_file, quiz1, quiz2, quiz3);
  if Name = c_name then
    s_num := i;
  end;
end;
close(record_file);
end;
```

```
procedure Show_Record;
begin
  clrscr;
  writeln('*****');
  writeln('* Student Record *');
  writeln('*****');
  if status='T' then
    begin
      writeln('Name Quiz1 Quiz2 Quiz3');
      for i:= 1 to 30 do
        with Student[i] do
          if Name <> '' then
            begin
              write(Name, ' ');
              write(quiz1, ' ');
              write(quiz2, ' ');
              write(quiz3, ' ');
              writeln;
            end;
      writeln('');
      repeat
        writeln('Press <Q> To Back To Main Menu');
        Make_choice_chr
      until choice_chr in ['Q','q'];
      if choice_chr in ['Q','q'] then
        begin
          if login_choice = 'S' then
            Students_Choices_List
          else
            Teachers_Choices_List
        end;
    end;
end;
```

```

end
else
begin
writeln('Name Quiz1 Quiz2 Quiz3');
with Student[s_num] do
begin
  write(Name,' ');
  write(quiz1,' ');
  write(quiz2,' ');
  write(quiz3);
  writeln;
end;
writeln('Press <Q> To Back To Main Menu');
Make_choice_chr;
if choice_chr in ['Q','q'] then
begin
  if login_choice ='S' then
    Students_Choices_List
  else
    Teachers_Choices_List
end;
end;

```

```

procedure Update_Record(quiz_done:char; score_record:integer);
begin
with student[s_num] do
  case quiz_done of
    '1': if quiz1 < 0 then quiz1 := score_record;
    '2': if quiz2 < 0 then quiz2 := score_record;
    '3': if quiz3 < 0 then quiz3 := score_record;
  end;
end;

```

```

procedure Update_Record_File;
var U_Record_file:text;
  U_Record_class:string;
begin
if status<>'T' then
  U_Record_class := Student_Class
else
  U_Record_class := Teacher_Class + ' ICT';

```

```

assign(U_record_file,HDD+'ICT_Learner\Records\' + U_Record_class +'.txt');
rewrite(U_record_file);
writeln(U_record_file, 'Name ');
writeln(U_record_file, 'Quiz1 Quiz2 Quiz3');
i:=0;
for i:= 1 to 30 do
begin
  with Student[i] do
    if Name <> '' then
      begin
        writeln(U_record_file,Name);
        writeln(U_record_file, quiz1 , ' ', quiz2 , ' ', quiz3 , ' ');
      end;
  end;
close(U_Record_file);
writeln('The record is updated!!!')
end;
{****-----End Of Record Panel-----****}

```

```

{*****
*  Comment Panel *
*****}

```

```

procedure Comment_Panel;
var comment_file:text;
  comments,comment_add,commentclass:string;
begin
  Clrscr;
  writeln('*****');
  writeln('* Comment Panel *');
  writeln('*****');
  if login_choice = 'S' then
    commentclass:=Student_Class
  else
    commentclass:=Teacher_Class+' ICT';
Assign(comment_file,HDD+'ICT_Learner\Comment\' +commentclass+'.TXT');
reset(comment_file);
while not eof(comment_file) do
begin
  readln(comment_file,comments);
  writeln(comments)

```

```

end;
Close(comment_file);

Assign(comment_file,HDD+'\ICT_Learner\Comment\' +commentclass+'.TXT');
Append(comment_file);
repeat
writeln('What comment you want to add?(Type <Q> to Exit)');
readln(comment_add);
if not ((comment_add ='Q')or(comment_add ='q')or (comment_add ='')) then
begin
writeln(comment_file);
Write(comment_file,commentclass+' '+c_name+: '+' comment_add);
end
else
if login_choice = 'S' then
begin
Close(comment_file);
Students_Choices_List
end
else
begin
Close(comment_file);
Teachers_Choices_List;
end;
until (comment_add ='Q')or(comment_add ='q');
end;
{*****--End Of Comment Panel Procedure---****}

```

```

{*****
*      Games Panel    *
*****}

```

```

{-----Vocabulary Guessing Game-----}
procedure lists_of_topic_of_game;
var game_topics: text;
game_chap: string;
counttp:integer;
begin
Clrscr;
assign(game_topics,HDD+'\ICT_Learner\Games\Vocab_game\secret_topic_menu.txt');
reset(game_topics);
counttp:=0 ;

```

```

while not eof(game_topics) do
begin
  readln(game_topics,game_chap);
  writeln(game_chap);
  counttp:=counttp+1
end;
writeln(counttp,' :Game Menu');
choiceL := counttp;
close(game_topics);
end;

procedure VocabGuessGame;
var
  Secret, Solution,topic_num ,chosenletters: string;
  Done : boolean;
  Letter : char;
  i : integer;
  secretfile : text;
  NoOfAttempts, NoOfWord : integer;
  word : array[1..50] of string;
begin
{----Select_topic----}
  lists_of_topic_of_game;
  repeat
    Make_choice;
    str(choice,topic_num);
    if choice = choiceL then
      Games_Menu;
    until (choice>= 1) and (choice <= choiceL) ;
{----SelectSecret----}
  assign(secretfile, HDD+'ICT_Learner\Games\Vocab_game\Vocab'+topic_num+'.txt');
  reset(secretfile);
  i := 0;
  while not eof(secretfile) do
begin
  i := i + 1;
  readln( secretfile, word[i])
end;
  close(secretfile);
  NoOfWord := i;
  randomize;
  Secret := Word[random(NoOfWord)+1];

```

```

{----Initialise----}
Solution := '';
for i := 1 to length(Secret) do
  Solution := Solution + '*';
NoOfAttempts := 2 * length(Secret);
Done := false;
chosenletters:='';
Clrscr;
writeln('The secret word to be guessed is ',Solution);
writeln('No. of Attempts remaining = ',NoOfAttempts) ;
writeln;
{----LetterGuess----}
repeat
begin
  write('The letter used: ');
  textColor(14);
  writeln(chosenletters);
  textColor(15);
  write('Please input the letter you guess(small letter): ');
  textColor(14);
  readln(letter);
  textColor(15);
  Clrscr;
  if ((ord(letter) >= 97) and (ord(letter)<= 122)) or (letter = ' ')then
  begin
    if Pos(letter,chosenletters) <= 0  then
      begin
        chosenletters := chosenletters + letter;
        for i := 1 to length(Secret) do
          if Letter = Secret[i] then
            Solution[i] := Letter;
        writeln('The secret word: ',Solution);
        NoOfAttempts := NoOfAttempts - 1;
        write('No. of Attempts remaining = ');
        textColor(14);
        writeln(NoOfAttempts);
        textColor(15);
      end
    else
      begin
        textColor(5);
        writeln('This letter has been chosen!!!');
      end
  end
end;

```

```
textcolor(15);
writeln('The secret word: ',Solution);
write('No. of Attempts remaining = ');
textcolor(14);
writeln(NoAttempts);
textcolor(15);
end
end
else
begin
textcolor(5);
writeln('Please Enter A Letter(a-z)!!!!');
textcolor(15);
writeln('The secret word: ',Solution);
write('No. of Attempts remaining = ');
textcolor(14);
writeln(NoAttempts);
textcolor(15);
end;
```

{----WinCheck----}

```
Done := (Solution = Secret) or (NoAttempts = 0);
until Done = true;
if Solution = Secret then
begin
textcolor(4);
writeln('You win!!');
textcolor(15)
end
else
begin
textcolor(4);
writeln('You lose!!');
textcolor(15);
end;
writeln;
repeat
writeln('Press Q to back to Games_Menu');
Make_choice_chr;
until choice_chr in ['Q','q'];
Games_Menu;
```

```
end;  
{-----End of VocabGuessGame procedure-----}
```

```
{-----Card Matching Game-----}  
procedure Draw_cards(fill_card:integer);forward;  
procedure Clear_card(clear_card:char);forward;  
procedure Read_Card_Content;forward;  
procedure Write_Card_Content(card_num:integer); forward;  
procedure Card_Choice_area;forward;  
{~}  
procedure Draw_cards(fill_card:integer);  
Begin  
  setfillstyle(solidfill,yellow);  
  setcolor(yellow);  
  case fill_card of  
    1:Bar(30,30,270,180);  
    2:Bar(290,30,530,180);  
    3:Bar(550,30,790,180);  
    4:Bar(30,200,270,350);  
    5:Bar(290,200,530,350);  
    6:Bar(550,200,790,350);  
    7:Bar(30,360,270,510);  
    8:Bar(290,360,530,510);  
    9:Bar(550,360,790,510);  
  end;  
  setcolor(9);  
  SetTextStyle(BoldFont, HorizDir,5);  
  case fill_card of  
    1:OutTextXY(140,90,'1');  
    2:OutTextXY(400,90,'2');  
    3:OutTextXY(660,90,'3');  
    4:OutTextXY(140,250,'4');  
    5:OutTextXY(400,250,'5');  
    6:OutTextXY(660,250,'6');  
    7:OutTextXY(140,410,'7');  
    8:OutTextXY(400,410,'8');  
    9:OutTextXY(660,410,'9');  
  end;  
End;  
procedure Clear_card(clear_card:char);  
begin  
  setfillstyle(emptyfill,0);
```

```

case clear_card of
  '1':Bar(30,30,270,180);
  '2':Bar(290,30,530,180);
  '3':Bar(550,30,790,180);
  '4':Bar(30,200,270,350);
  '5':Bar(290,200,530,350);
  '6':Bar(550,200,790,350);
  '7':Bar(30,360,270,510);
  '8':Bar(290,360,530,510);
  '9':Bar(550,360,790,510);
end;
end;

procedure Read_Card_Content;
var cardset_file:text;
  set_num:string;
begin
  Clrscr;
  writeln('Card Set Menu');
  writeln('1: Set 1');
  writeln('2: Set 2');
  writeln('3: Set 3');
  writeln('4: Game Menu');
repeat
  write('Which Set You Want To Play? ');
  readln(set_num);
until (set_num >= '1') and (set_num <= '4') ;
if set_num = '4' then
  Games_Menu;
assign(cardset_file,HDD+'\ICT_Learner\Games\Matching Game\' + 'CardSet' + set_num+'.txt');
reset(cardset_file);
readln(cardset_file, num_of_card);
num_of_pairs:= num_of_card div 2;
for i:= 1 to num_of_pairs do
  readln(cardset_file,answer_match[i]);
for i:= 1 to num_of_card do
begin
  for j:= 1 to 8 do
    readln(cardset_file,card_content[i,j]);
  readln(cardset_file);
end;
close(cardset_file);
end;

```

```

{-----}
procedure Write_Card_Content(card_num:integer);
  var x,y:integer;
begin
  case card_num of
    1:begin x:=30;y:=30; end;
    2:begin x:=290;y:=30; end;
    3:begin x:=550;y:=30;end;
    4:begin x:=30;y:=200; end;
    5:begin x:=290;y:=200; end;
    6:begin x:=550;y:=200; end;
    7:begin x:=30;y:=360; end;
    8:begin x:=290;y:=360;end;
    9:begin x:=550;y:=360;end;
  end;
  for i:= 1 to 8 do
    begin
      outtextxy(x,y,card_content[card_num,i]);
      y:=y+20;
    end;
  end;
{-----}
procedure Card_Choice_area;
var cardchoicelst,cardchoice2nd:char;
  user_match,user_match2 :string;

begin
  setfillstyle(emptyfill,0);
  Bar(30,530,600,630);
  moveto(30,530);
  setcolor(white);
  SetTextStyle(DefaultFont, HorizDir,1);
  {First card to filp}
  repeat
    setfillstyle(emptyfill,0);
    bar(0,530,2000,540);
    outtextxy(30,530,'The 1st card you want to flip:');
    cardchoicelst:= readkey;
    if not (cardchoicelst >= '1') and (cardchoicelst <= chr(num_of_card+48))
      then
        begin
          Bar(30,530,600,630);

```

```

outtextxy(30,530,'Please type a choice with 1 - ' + chr(num_of_card+48));
delay(1000)
end;

until (cardchoicelst >= '1') and (cardchoicelst <= chr(num_of_card+48));
outtextxy(275,530,cardchoicelst);
Clear_card(cardchoicelst);
Write_Card_Content(Ord(cardchoicelst)-48);
{Second card to flip}
repeat
  setfillstyle(emptyfill,0);
  bar(0,550,2000,560);
  outtextxy(30,550,'The 2nd card you want to flip:');
  cardchoice2nd:= readkey;
  if not (cardchoice2nd >= '1') and (cardchoice2nd <= chr(num_of_card+48))
    then
      begin
        Bar(30,550,2000,560);
        outtextxy(30,550,'Please type a choice with 1 - ' + chr(num_of_card+48));
        delay(1000)
      end;
  if   cardchoice2nd = cardchoicelst
    then
      begin
        Bar(30,550,2000,560);
        outtextxy(30,550,'Please type the other choice');
        delay(1000)
      end;
until (cardchoice2nd >= '1') and (cardchoice2nd <= chr(num_of_card+48))
  and (cardchoice2nd <> cardchoicelst);
outtextxy(275,550,cardchoice2nd);
Clear_card(cardchoice2nd);
write_Card_Content(Ord(cardchoice2nd)-48);
user_match:=cardchoicelst+'-' +cardchoice2nd;
user_match2:=cardchoice2nd+'-' + cardchoicelst;
i:=0;
repeat
  i:=i+1;
  if (user_match = answer_match[i]) or(user_match2 = answer_match[i]) then
    begin
      outtextxy(30,570,'You Have The Correct Match!');
      correct_match:=correct_match+1;
      fliped_times := fliped_times+1;
    end;

```

```

outtextxy(30,590,'You Tried '+ chr(fliped_times+48)+' Time(s)');
outtextxy(30,610,'You Have Got ' + chr(correct_match+48)+' Pair(s)');
Delay(1000);
end;
until (i = num_of_pairs) or (user_match = answer_match[i])
or(user_match2 = answer_match[i]);
if (i = num_of_pairs) and (user_match <> answer_match[i])
and(user_match2 <> answer_match[i]) then
begin
  outtextxy(30,570,'You Have A Wrong Match!');
  fliped_times:=fliped_times+1;
  outtextxy(30,590,'You Tried '+chr(fliped_times+48)+' Time(s)');
  outtextxy(30,610,'You Have Got ' + chr(correct_match+48)+' Pair(s)');
  delay(1500);
  Draw_cards(Ord(cardchoice1st)-48);
  Draw_cards(Ord(cardchoice2nd)-48);
end;
end;

procedure Matching_Game;
begin
  Read_Card_Content;
  correct_match:=0;
  fliped_times:=0;
  Writeln('Entering The Game, please wait...');
  GraphicsDriver := Detect;
  InitGraph(GraphicsDriver, GraphicsMode, '');
  for i:= 1 to num_of_card do
    Draw_cards(i);
  repeat
    Card_Choice_area;
  until correct_match = num_of_pairs;
  setfillstyle(emptyfill,0);
  Bar(30,530,2000,650);
  SetTextStyle(BoldFont, HorizDir,4);
  SetColor(3);
  outtextxy(140,570,'You Have Got All Pairs');
  outtextxy(30,650,'Back to Main Menu in 5 Seconds');
  delay(5000);
  closegraph;
  if login_choice = 'S' then
    Students_Choices_List
  else

```

```

Teachers_Choices_List;
end;
{-----End Of Card Matching Game Procedure-----}

{**----Fishing Game----**}
procedure Draw_Fish;
var Fish_file:text;
  Fish_code,Fish_line:string;
  Fish_code_temp:integer;
begin
  Clrscr;
  Randomize;
  Fish_code_temp:=random(6)+1;
  str(Fish_code_temp,Fish_code);
  assign(Fish_file,HDD+'\ICT_Learner\Games\Fishing Game\Fishes\Fish'+Fish_code+'.txt');
  reset(Fish_file);
  writeln;
  Randomize;
  TextColor(random(14)+2);
  while not eof(Fish_file) do
    begin
      write('          ');
      readln(Fish_file,Fish_line);
      writeln(Fish_line);
    end;
  for i:= 1 to 3 do writeln;
  close(Fish_file);
end;

procedure Fishing_Game;
var location_num:string;
  Fish_file:text;
  num_of_fish,FK,FC,FL,templocation_num:integer;
  Selected_bait:string;
  Which_bait,Available_bait,Wanted_bait :array[1..5] of string;
begin
  Clrscr;
  writeln('Location Menu');
  writeln('1: Location 1');
  writeln('2: Location 2');
  writeln('3: Location 3');
  writeln('4: Game Menu');

```

```

repeat
write('Which Location You Want To Fish? ');
readln(templocation_num);
until (templocation_num>= 1) and (templocation_num <= 4) ;
if templocation_num = 4 then
    Games_Menu;
str(templocation_num,location_num);
assign(Fish_file,HDD+'ICT_Learner\Games\Fishing Game\Location\Location'+ location_num+'.txt');
reset(Fish_file);
readln(Fish_file,num_of_fish);
for FL:= 1 to num_of_fish do
begin
    readln(Fish_file,Which_bait[FL]);
    readln(Fish_file,Available_bait[FL]);
    readln(Fish_file,Wanted_bait[FL])
end;
close(Fish_file);
writeln('There are ', num_of_fish, ' fish(es) in this location');
writeln;
FC:=0;
for FK:= 1 to num_of_fish do
begin
    Draw_Fish;
    TextColor(White);
    writeln('Fish ',FK);
    writeln(Which_bait[FK]);
    writeln(Available_bait[FK]);
    repeat
        write('Answer ? ');
        readln(Selected_bait);
        until Selected_bait<>;
    if (upcase(Selected_bait) = upcase(Wanted_bait[FK])) or
(Selected_bait = Wanted_bait[FK])then
begin
    writeln('You Caught The Fish!');
    writeln;
    FC := FC + 1
end
else
begin
    writeln('You Missed The Fish!');
    writeln('The Fish Wanted Bait ',Wanted_bait[FK]);
end

```

```

writeln;
end;
delay(500);
end;
writeln('You have caught ', FC, ' fish(es)');
writeln;
repeat
writeln('Press Q to back to Game Menu');
Make_choice_chr;
until choice_chr in['Q','q'];
if choice_chr in['Q','q'] then
Games_Menu;
end;
{**----End Of Fishing Game Procedure**}

```

{**-----Car Racing Game-----**}

```

procedure Draw_Car_u(A:integer;B:integer);forward;
procedure Draw_Car_c(A:integer;B:integer);forward;
procedure Clear_Track;forward;
procedure Read_Track_Content; forward;
procedure Move_Car_area;forward;

procedure Draw_Car_u(A:integer;B:integer);

```

```

Begin
setfillstyle(solidfill,Green);
setcolor(Green);
Bar(A,B,A+40,B+10);
End;

```

```
procedure Draw_Car_c(A:integer;B:integer);
```

```

Begin
setfillstyle(solidfill,yellow);
setcolor(yellow);
Bar(A,B,A+40,B+10);

```

```

End;
procedure Clear_Track;
begin
setfillstyle(emptyfill,0);
Bar(0,0,2700,230);

```

```

setfillstyle(solidfill,11);
Bar(30,30,110,200);
Bar(600,30,680,200);
SetTextStyle(DefaultFont,VertDir,1);
SetColor(6);
outtextxy(70,60,'Starting Line');
outtextxy(640,60,'Finishing Line');
end;

procedure Read_Track_Content;
var track_file:text;
track_num:string;
begin
Clrscr;
writeln('Track Menu');
writeln('1: Track 1');
writeln('2: Track 2');
writeln('3: Track 3');
writeln('4: Game Menu');
repeat
write('Which Track You Want To Race? ');
readln(track_num);
until (track_num >= '1') and (track_num <= '4') ;
if track_num = '4' then
  Games_Menu;
assign(track_file,HDD+'\ICT_Learner\Games\Car Racing Game\'+ 'TrackSet' + track_num+'.txt');
reset(track_file);
readln(track_file,CN);
for CL := 1 to CN do
begin
  readln(track_file,CQ[CL]);
  readln(track_file,CCH[CL]);
  readln(track_file,CANS[CL])
end;
close(track_file);
end;

{-----}
procedure Move_Car_area;
var User_Move:char;
ccpx,ccpy,ucpx,ucpy:integer;

```

```

begin
ccpx:=110;
ccpy:=40;
ucpx:=110;
ucpy:=80;
CNC:=0;
Clear_Track;
Draw_Car_u(ucpx,ucpy);
Draw_Car_c(ccpx,ccpy);
setcolor(white);
SetTextStyle(DefaultFont, HorizDir,1);
outtextxy(30,250,'There are '+ chr(CN+48) + ' Lap(s) To Run In This Competition');
Delay(3000);
for CK:= 1 to CN do
begin
  setfillstyle(emptyfill,0);
  Bar(30,250,2000,7000);
  moveto(30,250);
  setcolor(white);
  SetTextStyle(DefaultFont, HorizDir,1);
  outtext('Lap ');
  outtextxy(60,250,chr(CK+48));
  outtextxy(30,270,CQ[CK]);
  outtextxy(30,290,CCH[CK]);
  outtextxy(30,310,'Answer ? ');
  User_Move:=readkey;
  outtextxy(100,310,User_Move);
  if (User_Move = CANS[CK])or (upcase(User_Move) = upcase(CANS[CK])) then
    begin
      outtextxy(30,320,'Right!');
      CNC := CNC + 1;
      ucpx:=ucpx+(490 div CN);
      ccpx:=ccpx+(490 div CN);
    end
  else
    begin
      outtextxy(30,320,'Wrong!');
      outtextxy(30,340,'You Should Move To '+CANS[CK]);
      ccpx:=ccpx+(490 div CN);
    end;
  Clear_Track;
end;

```

```

Draw_Car_u(ucpx,ucpy);
Draw_Car_c(ccpx,ccpy);
end;
if ucpx = ccpx then
begin
  SetTextStyle(DefaultFont,HorizDir,2);
  Setcolor(10);
  outtextxy(30,360,'You Have Win The Match');
end
else
begin
  SetTextStyle(DefaultFont,HorizDir,2);
  Setcolor(10);
  outtextxy(30,360,'You Have Lose The Match');
end;

end;
procedure Car_Racing_Game;
begin
  Read_Track_Content;
  Writeln('Entering The Game, please wait...');

  GraphicsDriver := Detect;
  InitGraph(GraphicsDriver, GraphicsMode,'');

  Move_Car_area;
  setfillstyle(emptyfill,0);
  Bar(30,530,2000,650);
  SetTextStyle(BoldFont, HorizDir,4);
  SetColor(3);
  outtextxy(140,570,'You Have Finish The Match');
  outtextxy(30,650,'Back to Main Menu in 5 Seconds');
  delay(5000);
  closegraph;
  if login_choice = 'S' then
    Students_Choices_List
  else
    Teachers_Choices_List;
end;
{**-----End Of Car Racing Procedure-----**}

{*****}
*      Main Program      *
*****}

```

```

Begin
Welcome_Page;
writeln('LETS START LEARNING ICT');
writeln('ENJOY');
writeln('Loading...');

Delay(1000);
GetDriveName;
Delay(500);
Login;
Update_Record_File;
Good_Bye_Page;
End.

```

{*****

- * This Program Is Designed and Written By:
- * Cheung Sha Wan Catholic Secondary School
- * Chung Ho Yin 6A 7 2014-2015
- * Supervising Teacher:Mr. Chu
- * Final Edition Date:20-12-2014

***** }

The Whole program is uploaded to Dropbox:



https://www.dropbox.com/s/973ivva9sh24dv2/ICT_Learner.zip?dl=0

The old version is deleted and the updated version is uploaded for users to download



Appendix 2 - Working schedule

Date	Event
March-2014	Choice of Topic
April-2014	Background research + Define the objectives
1 st week of April	Search on Internet for the details of making an education system
2 nd week of April	Analysis the Information from Internet + Define the name of the system
3 rd week of April	Search the library for related books
4 nd week of April - 2 nd week of May	Read the books and jot down the related information.
3 rd week – 4 th week of May	Analysis the key points from books and the information from internet
June-2014 – July-2014	Design of Solution
June	Design the user interface and the data flow
July	Design the functions and the details
August-2014 –October-2014	Implementation
August	Implement the interface and functions
3-September	Ask teacher for some advice
September - 2 nd week of October	Finish almost the whole program with temporary files
3 rd week - 4 th week of October	Make all the resources files
November-2014 – December-2104	Testing & Evaluation
1 st week - 2 nd week of November	Internal testing and fix the found bugs
3 rd week of November - 1 st week of December	External testing and fix the found bugs
2 nd -3 rd week of December	Take a week of rest while making a user's manual, fine fixing the program and check the resources files
3 rd week of December	School Holiday: Go out and share my program with other schools' students to share our experiences
4 th week of December	Conclusion & Discussion + Final Report
1 st week of January	Check report ,the program and resources files Pack the program and the users' manual Make the report into a *.PDF
After the above things done	Hand in the project



Appendix 3 – User Manual

User Guide For ICT_Learner

System requirement	
Operating system	Windows 95/98/2000/me/XP
CPU	1GHZ or above
Hard disk memory space	10MB
Memory requirement	30MB RAM
Resolution	1024*768 or above
Hardware	Keyboard , Mouse , Monitor
Compiler	Dev-Pascal 7.0

*****Users should type the required choices for menus and every procedure.

*****Put the Folder ICT_Learner in the root of a hard disk (e.g. F:\)

Students:

1. Please make use of the program to learn.
2. Please type correctly for the menus, try not to type the invalid choices.

Teachers:

1. Please print the account and password for the students. (It is located at the 'account' folder)
2. The editing function is not very good:
Here are the structures of the files

Account:

Record:

Name
Quiz1 Quiz2 Quiz3
Thomas Chung
-1 -1 -1

Notes:

Try not to print over the screen.

Read the provided one for reference.

Exercises, quizzes, fishing game, car racing game:

3 ← Number of questions

Question 1

Choice

Answer

Question 2

Choice

Answer

Question 3

Choice

Answer

Maximum of 3 quizzes for the quiz panel.

Comment:

Make an empty line after the first one.

Vocab Game:

computer

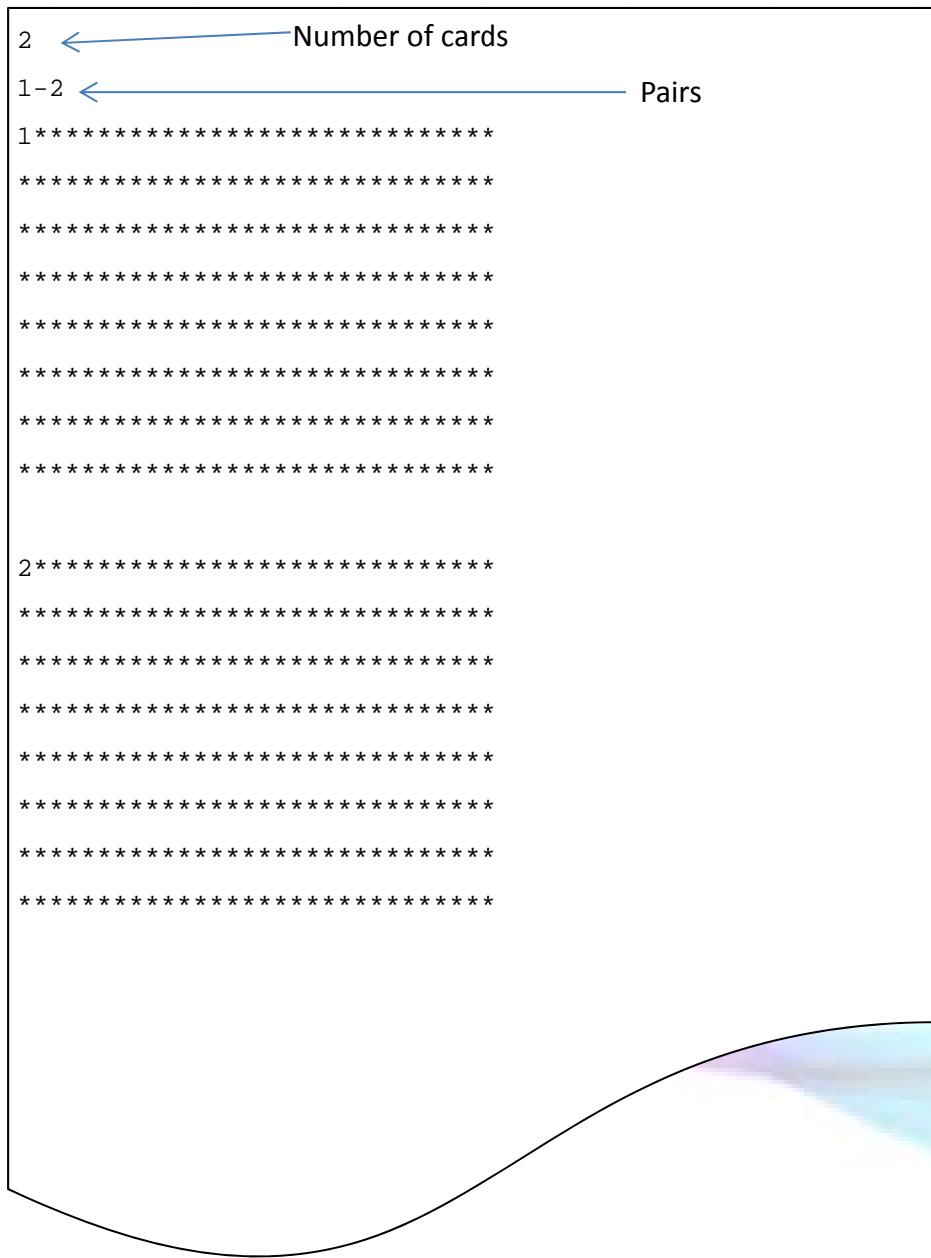
device

implement

Make the words in small letters is better.

Card game:

163



Just edit the like this, the stars in the middle is changed to word for game Type the correct pair after the one shown above when there are 4 cards

*****Number of pairs of card should be all the cards (At most one card can be left)*

*****See the provided files for reference.

Programmers or IT supporters:

Please send me an email to request the *.pas at thomas_chung1996@yahoo.com.hk

All the users are welcomed to report any bugs and inconvenience by sending me an email thomas_chung1996@yahoo.com.hk

The software publishing companies are also welcomed to contact by email

