Hong Kong Diploma of Secondary Education Examination

Information and Communication Technology
School-Based Assessment (SBA)

Module D: Software Development

Title: educational software program

Content

1) Introduction

1.1) Background Information

Nowadays, in Hong Kong, attract students' attention is one profound and significant way to let them focus on learning. It is because the learning environment in Hong Kong is making students feel antipathy and making their learning efficiency decrease. So that, there are many foreign countries such as US, Denmark, Germany, etc. are setting up many learning program to increase the students' learning interest. As the result, it can fund up a lot of students' learning interest in Hong Kong.

Moreover, the adolescent report shows that using game can attract more teens attention than the traditional ways. Some of the hobby groups are using games to help students to learn the content more quickly. Hence, we can see that the games are becoming one part for helping the students to learn.

1.2) Situation and the problems

As an IT project manager of a secondary school, the school would like to design an educational software program to assist students' learning. I am now responsible for the project. That's why I am working in this written report to provide solutions for the secondary.

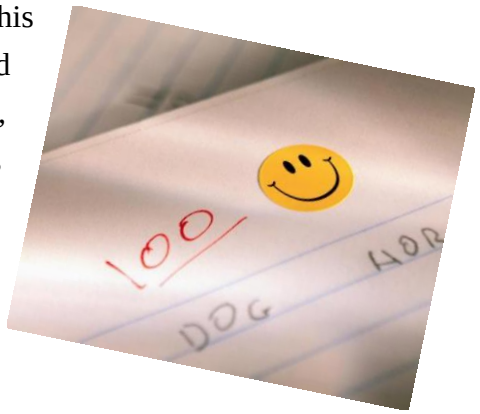
i) Purpose

- Boost children's incentive on learning

  In this system, we hope to use games to attract children to play this system that can throw this way to develop interest on different subject and have positive influence to their learning altitude.

- Let students to learn more useful knowledge on their core subject

  We have different games of different subject in this system such as the guess word game and crossword puzzle in English, arithmetic exercise in Mathematics, etc. We want to through different games, let the students to revise what their have learn in class and learn some extra knowledge which make them to improve their learning efficiency.



- Enhance the proficiency of the students

  The most important things about this system is want to increasing the proficiency of the students. Nowadays, the students are not good at some subject. For example, the English vocabulary and grammar, LS concepts and special terms and arithmetic skills. We hope to use this system to increase their proficiency of different subject. So those are the purposes of this system.

ii) About the product

In this project, I am going to develop a several game in English, Mathematics and Liberal Studies which are the core subject in the secondary school syllabus. And all games in this educational system which contains more than 1 game. As the educational system being done by various ways, guess word, crossword, multiple choice, fill in the blanks and arithmetic exercises. This game development can bring out the educational benefits for the students that can make them to learn there subject more easily.

iii) Target user' requirements and needs

1. **Target users**

As the system contains several games which are for help students to learn different subject in a fun ways. It this case, the focusing group of users will be junior forms students. This is going to make them can handle the secondary school syllabus more easily. Moreover, this software are also consist the teachers involve as it can let the teachers to edit the questions in the program if they needs.

But due to the difficulties of the questions and the development of the system, the senior forms students are not suitable in this program.

2. **Requirements and needs**

First of all, for the main users of this system, the system were considered whether the level of the game is suitable to them and all in syllabus as they are not old enough to handle the questions in the program, so that the system should be user-friendly and suitable for them to use.

Besides, for teachers, I think they can have the right to edit and create some of the questions as question should be change in a period of time to ensure less repeating questions would be done by students. Also, the teachers also have rights to observe the students' records as it uses to looks after the students' using

situation. So, it should be user-friendly as can ensure and make the teacher understand how the system works and how to use the system.

Moreover, for the administrator of this system, he/she can be make changes on the processing and procedures of the system to make it up to date of the system and provide better quality for the users.

iv) Main Functions

In this case, we need different functions in the program to cope with the requirements and needs.

1. **Using simply language of programming**

   It can reduce some complex controls of the system and processing so that students and teachers can use it easily and become more user-friendly.

2. **Adding login system**

   In this system, there will consist a lot of different users, the identical processing will be very difficult. As it is difficult to identify the identical of the users, a login function can help in this case, it will make the system become more comprehensive. Also, it is used to record the users' login data for some analysis. We can do it by create the account by the users. It can be ensure the ID of each user. In this function, if the users do not have their personal account, they can create a new one inside the system and they can edit their account in anytime they want. So that it can make the system become more user-friendly.

Other than those requirement and needs, more advanced functions are needed to make the system and program more interesting.

1. **Types of games**

   As I have mentioned before, instead of 1 type of game, I am going to produce several type of games to attract the users in different subjects. I plan to develop a guess words game, crossword puzzle, multiple choice questions and fill in the blanks.

Also, want to make the system become more diversification, I would like to add a simple arithmetic game. However, producing an extra game is time-consuming, it will need time to do it, and therefore the

time is a major point that I will explain it later.

2. Time controlling in the game

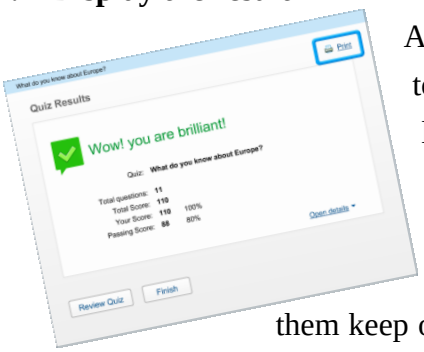
To make the system become more excited for the students, I will add the time recording in this game. The time recording is record the timing that how long the users are spent and show it after they are finish. It can make the game be more exciting and more efficient for them to know how much time are they spent.

3. Providing solutions about the game

The solutions should be given to the students as they can learn their mistakes from the answers. The answer can make them know what they are wrong and what they are miss. It is a profound element of each gaming system.

However, there are the risk when provide the solutions on the system. It is because there are not a large amount of questions within the system, there will be a risk that they will work on the same questions which they already knew the answers, so the teachers need to edit the questions in a period of time. Thus, a new function can be developed which can save several groups of questions at once reduce the sequence of editing questions.

4. Display the result



After playing the games and having fun, we need to get back to our major aim, improving their learning ability. So, the record of the students' score should be show. It can make the students and teachers can see their results and easily to make it be the reference. Hence, it can make them keep on to play and learn in our system, to help them to improve.

As it is an educational system, we the observer such as teachers need to know about what they are disadvantages. Hence, at the same time, we need to add a function for users entering their educated level, and then it can be more easily to know about their situation and help them to improve. So, a question should be asked before they create the account.

5. An attractive user page

In the game development, the interface will be most important things in each game. The attractive user page can make the system become more famous and can attract more students interest. In this system, I have made some different colorful interface in this system, it can help to provide better learning environment and can make the system become attractive for the students.

2) Analysis

2.1) Preliminary Study

As before doing the learning system, the research should be done to make the myself to have the well-prepared and basic concept for the project.

i) Data research

From the Internet, there are different types of games in different subject. We can find some of the sample in there to be the reference. However, the major function of those from the Internet has a big different than I thought that those are much more advanced of the programming system.

But the observation show that, the common features of the system that they are full of color which are the attractive users' interface. Also, many of the gaming system are also hard to be developed as they always contain some animations that Pascal language cannot support.

Besides, some functions which I think it can help my system are to provide the users records, login function, etc. It can help the system become more comprehensive. So the observation is the most helpful ways for me to develop this system.

ii) Interview

It is important to de the preparation before producing the program as the users of this program are the public and their views can affect the functions of the program. We need to base on their needs to set up a system for them. Therefore, I would like to conduct an interview to ask for public ideas to be the reference of the program.

So, after collecting the data and analyzing them, the data can help me to develop the program. As the result, the data can help the program fit in with

the public views.

Here is the collected data from the users:

	Students	Teachers	Administrator
Login function	✓	✓	✓
Different types of game	✓	✓	
Display the result	✓	✓	
Opinion collection			✓

iii) Data analysis

From the above table, all the stakeholders (students, teachers, and administrator) are think that the login function are very important in this system, because it can help to prevent losing the data and the personal information such as the login ID, password, or even the personal record. It can protect their personal account and easily to observe their record by teachers. Hence, all the stakeholders would like to have this function in this program.

Besides, the users' opinions are the profound resources to make the program improve. So the administrator would like to receive the opinion for us to improve as at the same time, we can provide better quality of services to the users.

About the display record, teachers and students would like to have the records display that because it can make the students know about how much scores they are having and the teachers can more easily to observe their performance.

Moreover, the teachers and students are also wanted to have more different types of games in this system. About the students, it can have more fun and choice in this program. About the teachers, it can collect the performance of students in different perspectives. However, as there are more type of games, teachers and administrator will bear a heavy workload because

teachers needs to prepare different questions and answers for the students, also it will need an analysis for different games, which means the workload is very heavy for the teachers to bear.

3) Design

3.1) Description

As the aim of this system is to provide a comprehensive education game system of the core subject (English, Mathematics, Liberal Studies), I am going to set up the game system with different functions.

In this program, it should not only include some games in it, the system should also include some other functions like login system, scores display, etc. So the program can be comprehensive enough and reach the real situation of a game system.

To make the system better in image, I use different color on the font and the background so that it seems to be more attractive. Followed by it is a login function and a creative account function. When the users login to system, they will see the main menu with 6 different columns (the arrangement is English game, Mathematics game, Liberal Studies game, Scores Display and Logout) so that the programs can be well-arranged.

For the scores display, the screen will show the information of the student which will show the user' score, name and the year of forms. The table will be arrange probably that to show the students' information.

Each game will consist of a unique instruction so that the users can know clearly about what they are going to play and what they are going to play. When a round of game is over, the users can choose to continue or leave the page so that it will be more convenient for the users to play the game continuously.

◆ Functional Design

3.2) Proposed Functions

Based on the above data, several functions must be produced, and they are login function, types of game, display result and opinion and opinion collection.

i) Types of game

In this system, there are 5 games are produced. I am going to produce crosswords puzzle, guess-word in English; multiple choice questions, fill in the blanks in Liberal Studies; arithmetic practice and challenge in Mathematics.

a) English Game:

Guess-words --- There is several questions are produced for students to answer. And the computer will choose one question randomly in each time.

Crosswords puzzle --- This is a traditional crosswords puzzle for students to play. The format of this puzzle will be 5x5. And the computer will also choose 1 set of question out of 4 randomly for students to answer.

b) Mathematic Game:

Arithmetic practice --- The practice is containing 3 basic arithmetic practices for students (addition, subtraction and multiplication). Each practice is having 10 questions and it is produce by computer randomly.

Challenge mode --- The challenge mode are same as the practice that it also contain 10 questions and draw randomly. But the questions are mess of 3 types of arithmetic calculation and students need to follow the arithmetic rule to answer the question.

c) **Liberal Studies Game:**

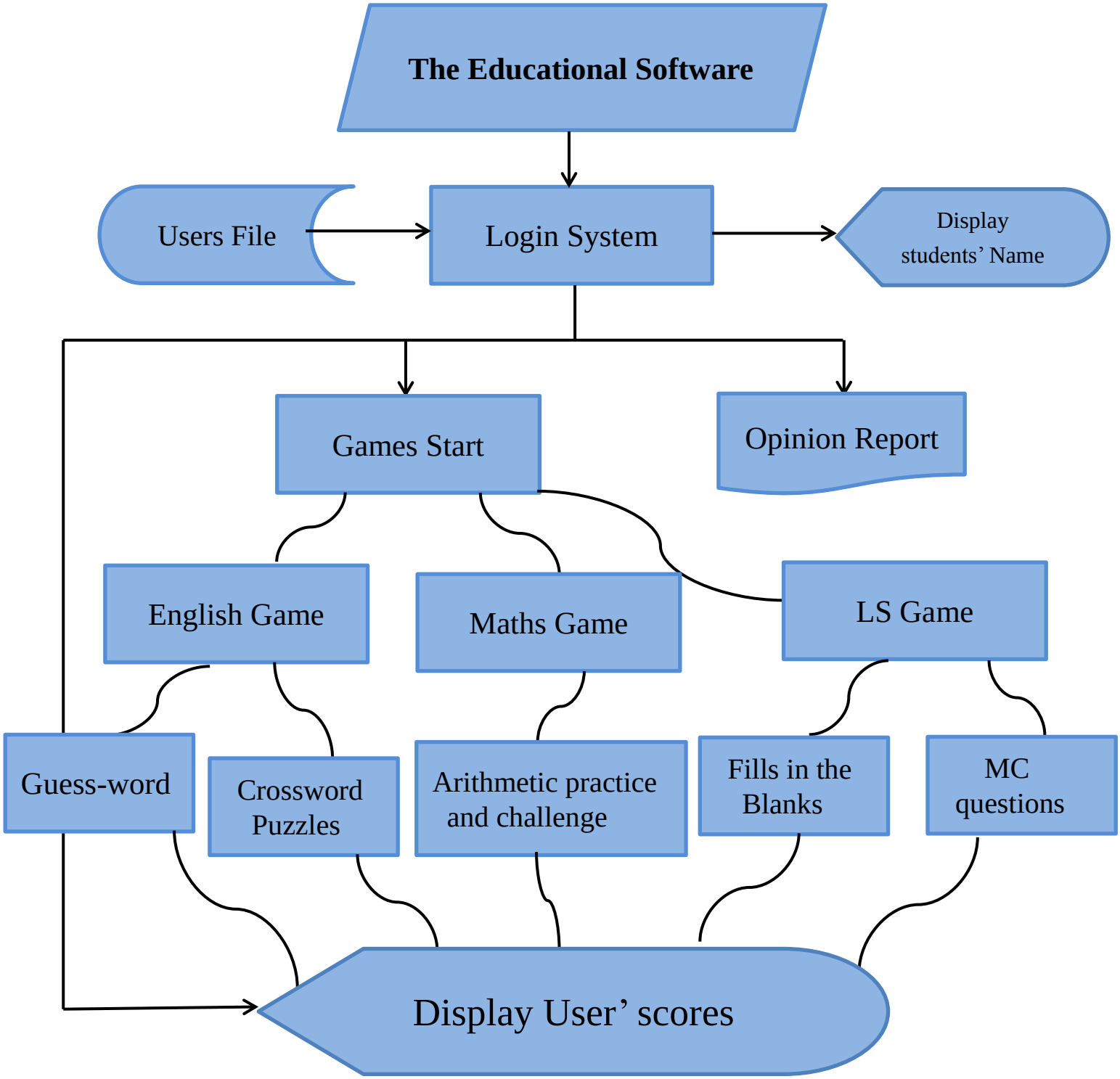
Multiple choice --- There are 3 choices for each question, A, B, C. And there are 15 questions in each set of multiple choice that draw by the computer randomly.

Fill in the blanks --- There are a blank in each sentence. The answer may be a special terms or some vocabulary. The questions are drawing randomly from the computer.

◆ System Design

3.3) The Construction of program

System Flowchart



i) Login system

The main function is using to identify the identity of Users and ensure the security of the program.

Input

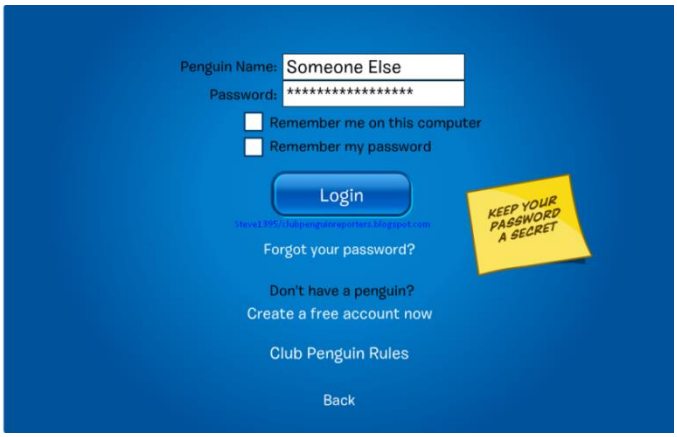
The user should enter their User-Name and password when they want to login the system. If the users do not have their personal account, they can create a new one inside the system and they can edit their account in anytime they want.

Process

When they enter their personal account, the system will check whether the data is valid or not by checking their entered information is or not same as their application before. For example, if their entered user-name and password are “abcdef” and “123456”, but their application users-name and password are “abcdf” and “123457”, then the system will not continue.

About to create account, the system will check there are or not have the same user-name or password already.

About to edit the account, the system will find out the account which is same as the input by the user who wants to edit.



Output

A successful or unsuccessful login will be shown in the screen. If it is success, the system will continue. If it is an unsuccessful login, there will be 3 functions, one is re-login, other one is forgot password. Forgot password can provide one opinions when forgot the password, such as find the administrator. Another one is for students to create a new account.

For the create account, If there has the existed account, then the system will not accepted this account and tell the user to set up the other account which is not existed.

For the edition account, if the system can find the account which same as the input then the user can be change it will the input is correct.

Data Storage

We need to store the login data as there will be records many data while the users are login and to store the account which is new and the new edition. Also it is used for saving the score of the users. Hence, we will save the data in a text file.

ii) Opinion Collection

It is used to collect users' opinions about the program and stoe the opinions in a specific file.

Input

When the users choose to give opinion, they can enter the opinions ehich he wanted to give about the game.

Process

When the users summit their opinion, the text which the users entered will automatically save in a text file and wait for the administrator to check for it.

Output

After the user entered the opinions about the system, the screen will be return to the menu.

iii) Display Scores

It is used to check the scores of the user after they are play the game.

Output

Scores of the users who after finish the game which be shown in the screen with their user-name, English Names, Form of years.

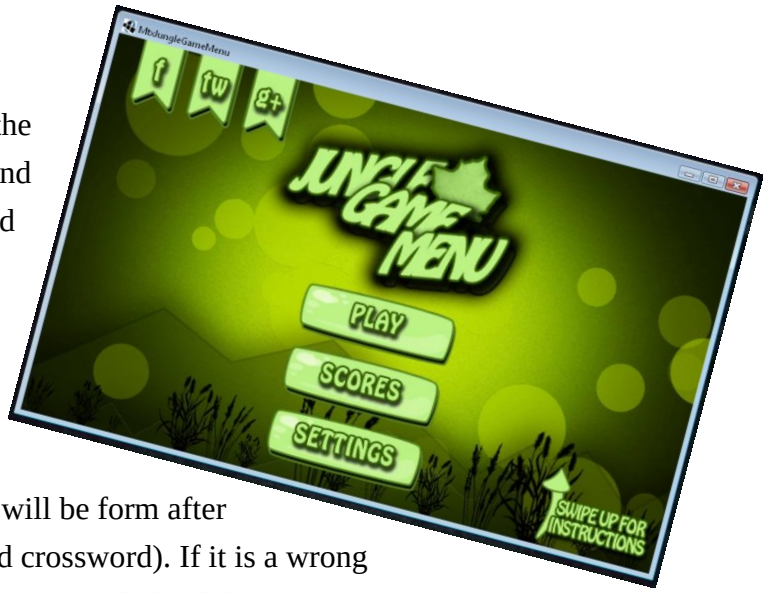
iv) Games

Input

When the users enter of the games, they will start to play the games and answer the questions. In this system, the user needs to answer different questions in each different game. In English game, for the guess-word, the users need to follow the hints and enter different letters to fulfill the questions; for the crossword puzzle, the users also need to follow the hints and complete the table by enter the words. In Mathematics game, for arithmetic practice and challenge, the users need to enter the answer of 15 questions in addition, subtraction, multiplication and the mixture of those three type. In Liberal Studies games, for multiple choices, the users only need to enter the letter A-C simply to answer the question; for fill in the blank, they need to spell the whole special term with correct capital letter.

Process

The system will check the answers of different questions and calculate the numbers of correct and wrong answer. Also the scores will be add in this stage in their own account



Output

The correct or wrong statement will be form after they answer each question (excepted crossword). If it is a wrong answer, the correct answer will be appearing behind the wrong statement. After the finish the game, the result of the users will be shown in the screen.

After that, the scores will be change in the "Score Display" simultaneously.

3.4) Difficulties

About the difficulties of making this program, there are something problem are border the creation of the program.

i) Time available

As I mentioned in above (Main Function), the time perspective is one thing to increase the difficulties of the program. It is because time may make the system quality become low as the programmer cannot have enough time to debugs or increase the quality.

For me, the time limitation makes me hard to create a high quality game and to make the system become more interesting. Also, assign a time to the system is hard that lots of time is needed for building up each game in this system and some data research. So the time perspective is one difficulties to create this system.

ii) Limitation of the software

The limitation of the Pascal and relevant software will also increase the difficulties of the program. In this case, Pascal is one of the software that cannot provide a Chinese character and cannot provide a high quality interface. It will make this system hard to reach the original design of the programmer.

About Pascal cannot provides a good interface, it is the other difficulties to make me hard to create a program. As we know that, Pascal is one of the software that cannot provide a good interface, it makes the image of the system cannot reach my expectation. It will make some of the games cannot present in graphical method.

Moreover, about Pascal cannot provide a Chinese character, it make the system cannot produce some game with Chinese characters. In this system, I have an idea to create the Chinese game in this system in my original plan, but the Pascal cannot provide a Chinese character that

make the system cannot present the Chinese game. So, the limitation of the software will also increase the difficulties of making this program.

3.5) General Function of the System

Here I will have some short introduction about the design of the general function of this system:

i) Login Account function

To make the system become more reality and more closer to a real game system, I design the login system in this program to allow the users to login to the program to identify them.

In this login system, there has another usage which is to save scores of the particular users after they play the games.

The login function is also set for users to logout and let other users to use the system instead of re-opening the system again.

ii) Create Account Function

This system is available for users to create their own account and the system does not provide the built-in account for the users to use. In this function, the system can check the new created account whether it has existed already or not.

iii) Amend Account Function

The system will provide the amend account function for users to change their user-name or password which can make the function of the system can much friendly to the users and more closer to the Internet games' function.

iv) Opinion Collection Function

The opinion collection function has created in the system that to let the users to provide their opinions to the administrator as there will be have some bugs in the system. This function can help the administrator to improve the system.

3.6) Flow of the system

1) **Cover Page:** Welcome page

The system will show the welcome page which has “Learning gamer” logo in there.

2) **Login Selection:** Enable users to choose to login account, create account or to amend account.

- Login page:

The system will ask the user-name and password of the users. The system will let the user to login id the username and password is correct. If not, the page with two strings “Login again” and “create account” will appear.

- Create account:

The system will ask the user-name, password, English Name and form of year of the users. The system will allow the users to create account with name not same as the existed account. If not, the page with the string ”The account is existed” will appear.

- Amend account:

The system will ask the user-name. The system will allow the user to change their user-name and password if the user-name is correct. If not, the page with the string “The user-name is not correct” will appear.

3) **Main Menu:** Allow users to choose for different functions

i) English Game
   a) Enter the sub-menu of English Game
   b) Choice the game you want to play
   c) Return to the sub-menu
   d) Return to the main menu

ii) Mathematic Game
   a) Enter the sub-menu of Mathematic Game
   b) Choice the game you want to play
   c) Return to the sub-menu
   d) Return to the main menu

iii) Liberal Studies Game
   a) Enter the sub-menu of Liberal Studies Game
   b) Choice the game you want to play
   c) Return to the sub-menu
   d) Return to the main menu

iv) Check your scores
   a) Check the scores
   b) Return to the main menu

v) Logout

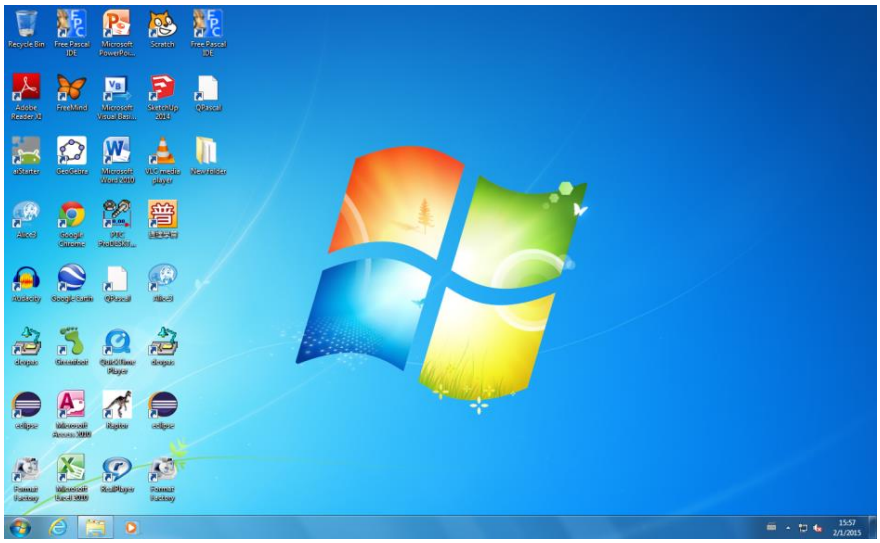
3.7) Choice of Operating System

I chose Windows OS to be the operating system because the operating system in that it is quite user-friendly and the DOS[1] environment of the system can only operate on Windows. Besides, Pascal (A full-featured IDE[2]) based on Windows, so Windows OS has chosen as my developing and program's operating system.

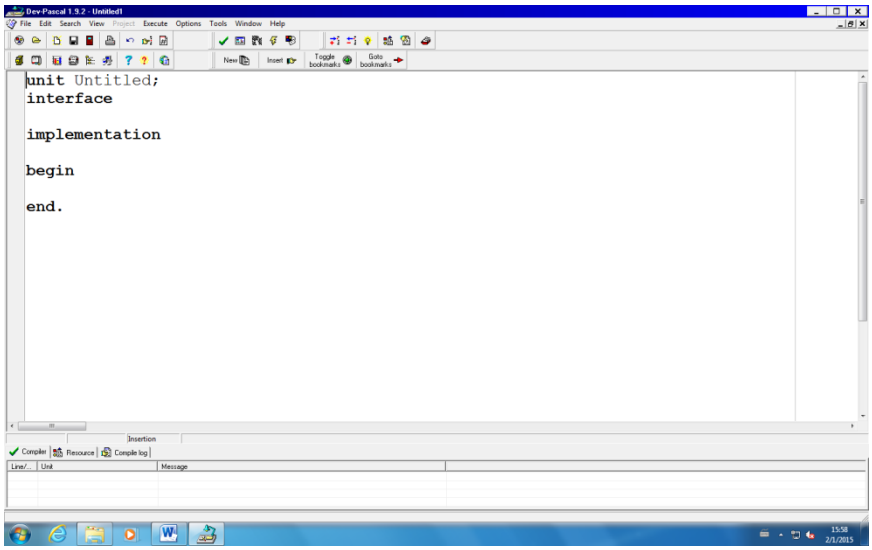
3.8) Software required

In this project, there are some software are used to build up this system. In the following, I will introduced the software are used in this project.

1. The Window OS (A platform to executing and developing the program)

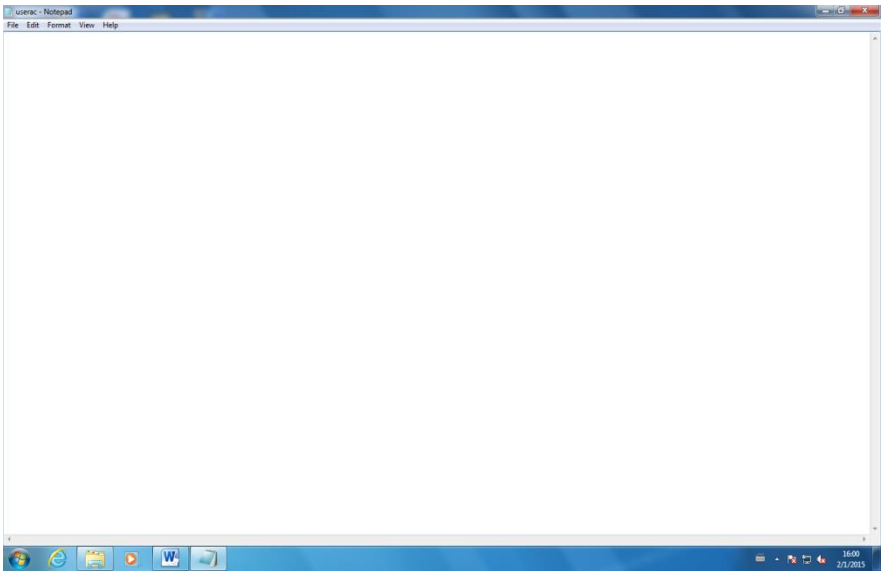


2. Dev-Pascal (The IDE of Program developing Software)



[1] DOS: Disk Operating System
[2] IDE : integrated development environment

3. Note-pad (The software to assist the developing software which use to store the information of the users, words, etc.)



4) Implementation

4.1) Description

To implement the system, the Dev-Pascal has chosen as mentioned in above. The source program has made to compile to an executable program.

First of all, in this program, I will design the simply structure of the program by considering the procedure in the program. About the procedure of the program, the sample outputs are well-considered as the system should be user-friendly and comprehensive for the user. So the program will use the different program codes, procedures, functions.

In this section, I will talk about the main program code of the program, and explain the screen layout of the system. So this part can help the user to have a deeper knowledge of the system.

4.2) Program code

First of all, a colorful welcome page will be shown.

```
begin
   ReadRecord;
   textcolor(4);
   writeln('    ============================================================================');
   writeln('    {  L       EEEEEEE     A     RRRRRR  NN    N IIIII NN    N   GGGGGG    }');
   writeln('    {  L       E          A A    R     R N N   N   I   N N   N  G          }');
   writeln('    {  L       EEEEEEE   AAAAA   RRRRRR  N  N  N   I   N  N  N G    GGGG  }');
   writeln('    {  L       E        A     A  RRRR    N   N N   I   N   N N  G      G   }');
   writeln('    {  LLLLLLL EEEEEEE A       A R   RRR N    NN IIIII N    NN   GGGGGG    }');
   textcolor(6);
   writeln('    {                                                                        }');
   writeln('    {             GGGGGG        A     M       M EEEEEEE RRRRRR             }');
   writeln('    {           G             A A    MM     MM E       R     R            }');
   writeln('    {          G      GGGG   AAAAA   M M   M M EEEEEEE RRRRRR             }');
   writeln('    {           G      G    A     A  M  M M  M E       RRRR               }');
   writeln('    {            GGGGGG    A       A M   M   M EEEEEEE R   RRR            }');
   textcolor(3);
   writeln('    {     ============================================================== }');
   writeln('    {Time: ',DateTimeToStr(now), '                                                  }');
   writeln('    {                                                                        }');
   writeln('    {                                                                        }');
   writeln('    {                                                                        }');
   writeln('    {                                                                        }');
   writeln('    {                     Please press ENTER to continue                     }');
   writeln('    ----------------------------------------------------------------------------');
   readln;
   UserRecords;
   Selectgame(index);
end.
```

After that a login menu will be shown.

```
    repeat
       textbackground(15);
       textcolor(0);
       clrscr;
       writeln;
       writeln;
       writeln;
       writeln('                      Welcome to play the learning gamer   ');
       writeln('                    Time: ',DateTimeToStr(now));
       writeln;
       textcolor(3);
       writeln('                    ======================================');
       writeln('                    *                                    *');
       writeln('                    *       1.  login your account       *');
       writeln('                    *                                    *');
       writeln('                    *       2.  Create your new account  *');
       writeln('                    *                                    *');
       writeln('                    *       3.  Update your account      *');
       writeln('                    *                                    *');
       writeln('                    *       4.  Quit                     *');
       writeln('                    *                                    *');
       writeln('                    ======================================');
       writeln;
       repeat
          write('                      Enter your choice (1-4): ');
          readln(choice)
       until choice in [1..4];
       case choice of
          1 : login(index);
          2 : CreateAccount;
          3 : UpdateAC(UserNum);
    end;
    until choice = 4;
end;
```

Each option will call different procedure, which means go to different parts of the login system. If the user enters “1”, the system will turn to the Login page. If the user enters “2”, the system will turn to the Create account page. If the user enters “3”, the system will turn to the Update account page. If the user enters“4”, the system will quit.

When enter”1”,

```
procedure login(var index : integer);
var
   userid, userpw     : string[30];
   found              : boolean;
   n, selection : integer;
begin
   repeat
      Clrscr;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      write('               Enter your user name: ');
      readln(userid);
      writeln;
      writeln;
      writeln;
      write('               Enter your password: ');
      userpw := GetPword;
      found := false;
      n := 0;
      repeat
         n := n + 1;
         with Player[n] do
            if (UserName = userid) and (Password = userpw) then
               begin
                  index := n;
                  found := true;
               end
      until found or (n > numuser);
```

```
      if found then
         begin
            Clrscr;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln('                          Hello! Welcome ', Player[index].EngName,'!');
            writeln;
            writeln;
            write('                            Press Enter to continue!');
            readln;
            Selectgame(index);
         end
      else
         begin
            Clrscr;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln('                           Wrong UserName or Password!');
            writeln;
            writeln('                          1. login again?');
            writeln;
            writeln;
            writeln;
            writeln('                          2. Don"t have account? Create one now');
            writeln;
            writeln;
            writeln;
            write('Your selection? ');
            readln(selection);
          end;
   until found or (selection = 2);
         if selection = 2 then
            CreateAccount
end;
```

When enters”2”,

```
procedure CreateAccount;
var
   m, index : integer;
   found    : boolean;
   target   : string[30];
begin
   Clrscr;
   assign(UserAccount, 'userac.txt');
   append(UserAccount);
   numuser := numuser + 1;
   with Player[numuser] do
   begin
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      write('              Enter your user name : ');
      readln(UserName);
      writeln;
      writeln;
      found := false;
      m := 0;
      repeat
         m := m + 1;
         with Player[m] do
            if UserName = target then
               begin
                  index := m;
                  found := true
               end
      until found or (m = numuser);
```

```
      if found then
         begin
            writeln('Invalid user name! Please input again');
            with Player[index] do
               repeat
                  begin
                     write('              Enter your user name : ');
                     readln(UserName);
                     writeln;
                     writeln;
                  end;
               until found = false;
         end;
      write('              Enter your password : ');
      Password := GetPWord;
      writeln;
      writeln;
      write('              Enter your English Name: ');
      readln(EngName);
      writeln;
      writeln;
      write('              Enter of Year of form (e.g. "F3", "F6") : ');
      readln(Form);
      writeln(UserAccount, UserName);
      writeln(UserAccount, Password);
      writeln(UserAccount, EngName);
      writeln(UserAccount, Form);
      writeln(UserAccount, 0);
      writeln(UserAccount, 0);
      close(UserAccount);
   end;
end;
```

When enters “3”,

```
procedure UpdateAC(var count : integer);
var
   target   : string[30];
   found    : boolean;
   n, index : integer;
begin
   Clrscr;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln('                              Update Your Account');
   writeln;
   writeln;
   write('                      Enter your User Name: ');
   readln(target);
   found := false;
   n := 0;
   repeat
      n := n + 1;
      with Player[n] do
         if UserName = target then
           begin
              index := n;
              found := true
           end
   until found or (n = numuser);
   if found then
     begin
```

```
            Clrscr;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            write('                                Finding player record...');
            delay(2000);
            with Player[index] do
            begin
               Clrscr;
               writeln;
               writeln;
               writeln;
               writeln;
               writeln;
               writeln;
               writeln;
               writeln('                                 Player record found!');
               writeln;
               writeln;
               write('                Enter your user name: ');
               readln(UserName);
               writeln;
               writeln;
               write('                Enter your password: ');
               GetPWord;
               writeln;
               writeln;
               writeln;
               writeln;
               store_users_info;
               writeln('                                    Record updated!');
               writeln('                                Please Enter to return.');
               readln;
               writeln;
               writeln;
            end
         end
      else
         begin
            Clrscr;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            write('                                Student name not found!');
            readln;
         end;
   end;
```

After login the system, the main menu will be shown.

```
procedure Selectgame(index : integer);
var
   choice : integer;
begin
   repeat
      textbackground(0);
      Clrscr;
      textcolor(15);
      writeln;
      writeln;
      writeln('                          Welcome to play the learning gamer   ');
      writeln('                       Time: ',DateTimeToStr(now));
      writeln;
      writeln('                       ========================================');
      writeln('                       *                                      *');
      writeln('                       *       1.  English Game               *');
      writeln('                       *                                      *');
      writeln('                       *       2.  Mathemetics Game           *');
      writeln('                       *                                      *');
      writeln('                       *       3.  LS Game                    *');
      writeln('                       *                                      *');
      writeln('                       *       4.  Your Score                 *');
      writeln('                       *                                      *');
      writeln('                       *       5.  Your Opinions              *');
      writeln('                       *                                      *');
      writeln('                       *       6.  Logout                     *');
      writeln('                       *                                      *');
      writeln('                       ****************************************');
      writeln;
      repeat
         write('                        Please enter your game choice: ');
         readln(choice);
      until choice in [1..6];
      case choice of
         1 : EnglishGame;
         2 : MathsGame;
         3 : LSGame;
         4 : display_user_score(index);
         5 : UserOpinions;
      end;
   until choice = 6
end;
```

In the main menu, there are the same type of function which for the users to call different parts of the program. If the user enters “1”, the system will turn to the English Game. If the user enters “2”, the system will turn to the Mathematics Game. If the user enters “3”, the system will turn to the Liberal Studies Game. If the user enters“4”, the system will turn to the Scores Display. If the user enters“5”, the system will turn to the Opinions collection. If the user enters“6”, the system will return to the login system.

For option “1” in the main menu,

```
begin
   repeat
   textbackground(14);
   textcolor(4);
   Clrscr;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln;
   writeln('                    Time: ',DateTimeToStr(now));
   writeln('                                English Game           ');
   writeln('                    ======================================');
   writeln('                    {                                    }');
   writeln('                    {      1. English Guess Game         }');
   writeln('                    {                                    }');
   writeln('                    {      2. Crossword Puzzle           }');
   writeln('                    {                                    }');
   writeln('                    {      3. Quit                       }');
   writeln('                    ======================================');
      repeat
         write('                         Enter your choice (1-3): ');
         readln(Selection);
      until Selection in [1..3];
      case Selection of
         1 : EngGuessGame;
         2 : CrossWPuzzle;
      end;
   until Selection = 3;
end;
```

The sub-menu about the English Game is shown to the user.

From the above graph, we can see that when user selects option”1”, the program will turn to the English Guess-word game

```
   procedure EngGuessGame;          {Start of Guessword game}
   var
      Secret, Solution, Hint_Strg, Meaning : string;
      Done : boolean;
      Letter : char;
      NoOfAttempts : integer;
      word : array[1..50] of string;
      hint : array[1..50] of string;
      mean : array[1..50] of string;
      NoOfWord : integer;
      Continue : char;

      procedure readwords;
      var
         i : integer;
         infile : text;
      begin
```

and selects option"2", the program will turn to Crossword game.

```
procedure CrossWPuzzle;        {Start of crossword puzzle}
const  max_no_q = 4;
var
   hint_num, quest_num, num_corr, num_char : integer;
   word, content : array[1..5, 1..5] of char;
   hint: array[1..25] of string;
   choice : string;
   valid_input, stop_game : boolean;
   try_again : char;
   quest_no_str : char;


   procedure Get_Words(quest_no : integer);
   var
      wordfile : text;
      i, j : integer;
   begin
      assign(wordfile, 'WORDS' + quest_no_str + '.txt');
      reset(wordfile);
```

The user can choose the game which they want to play and play it.

For option "2" in the main menu,

```
begin
   repeat
      Clrscr;
      textbackground(3);
      textcolor(7);
      Clrscr;
      writeln;
      writeln;
      writeln;
      writeln('                        Time: ',DateTimeToStr(now));
      writeln('                                 Maths_fun_game            ');
      writeln('                        ========================================');
      writeln('                        {                                      }');
      writeln('                        {      1. Addition Game                }');
      writeln('                        {                                      }');
      writeln('                        {      2. Subtraction Game             }');
      writeln('                        {                                      }');
      writeln('                        {      3. Multiplication Game          }');
      writeln('                        {                                      }');
      writeln('                        {      4. Challenge Mode               }');
      writeln('                        {                                      }');
      writeln('                        {      5. Quit                         }');
      writeln('                        {                                      }');
      writeln('                        {                                      }');
      writeln('                        ========================================');
      repeat
         write('                         Enter your choice (1-5): ');
         readln(choice)
      until choice in [1..5];
      case choice of
         1 : Addition_Game;
         2 : Subtraction_Game;
         3 : Multiplication_Game;
         4 : Challenge Mode;
      end;
    until choice = 5;
end;
```

The sub-menu of the Mathematics Game will be shown.

From the above graph, we can see that when user selects option"1", the program will turn to the Addition game.

```
procedure Addition_Game;
var
   NumOfCorrect, NoOfQuest, i : integer;
   num1, num2, num3, noNum, Ans : integer;
   Userans : integer;
   time : integer;
begin
   Clrscr;
   time:= gettickcount;
   NoOfQuest := 10;
   NumOfCorrect := 0;
   randomize;
```

If the students select option"2", the program will turn to Subtraction game.

```
procedure Subtraction_Game;
var
   NumOfCorrect, NoOfQuest, i : integer;
   num1, num2, num3, noNum, Ans : integer;
   Userans : integer;
   time : integer;
begin
   Clrscr;
   time:= gettickcount;
   NoOfQuest := 10;
   NumOfCorrect := 0;
   randomize;
```

When the students select option"3", the program will turn to Multiplication game

```
procedure Multiplication_Game;
var
   NumOfCorrect, NoOfQuest, i : integer;
   num1, num2, num3, noNum, Ans : integer;
   Userans : integer;
   time : integer;
begin
   Clrscr;
   time:= gettickcount;
   NoOfQuest := 10;
   NumOfCorrect := 0;
   randomize;
```

and the students select option"4", the program will turn to Challenge Mode.

```
procedure Challenge_Mode;
var
   NumOfCorrect, NoOfQuest, i : integer;
   num1, num2, num3, opcode1, opcode2, Ans : integer;
   Userans : integer;
   time : integer;

function Numresult(num1, num2, opcode : integer) : integer;
begin
{0 represents + , 1 represents - , 2 represents *}
   case opcode of
      0 : Numresult := num1 + num2;
      1 : Numresult := num1 - num2;
      2 : Numresult := num1 * num2;
   end;
end;
```

In this section, the user also can choose the game which they want to play and play it.

If choice “3” is selected by user in the main menu, procedure of Liberal Studies Game is called.

```
begin
   repeat
      textbackground(1);
      textcolor(7);
      Clrscr;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln('                    Time: ',DateTimeToStr(now));
      writeln('                              LS Foundation Game          ');
      writeln('                    ========================================');
      writeln('                    {                                      }');
      writeln('                    {      1. Multiple Choice Game         }');
      writeln('                    {                                      }');
      writeln('                    {      2. Fill in the blanks           }');
      writeln('                    {                                      }');
      writeln('                    {      3. Quit                         }');
      writeln('                    ========================================');
      repeat
         write('                     Enter your choice (1-3): ');
         readln(choice)
      until choice in [1..3];
      case choice of
         1 : LSMc;
         2 : LSFill_in_the_blanks;
      end;
   until choice = 3;
end;
```

The sub-menu of the Liberal Studies Game will be shown.

From the above graph, we can see that when user selects option”1”, the program will turn to the Multiple Choice Game.

```
procedure LSMc;      {LSMc Game}
var
   NoOfQuest, L, NumOfCorrect : integer;
   MCQuestions, ChoOfQuest : array[1..20] of string;
   ModelAns : array[1..20] of char;
   MCQuestAns : char;
   MCQuests, MCChoice : string;
   MCFile : text;


   procedure Read_MCFile;
   var
      L : integer;
   begin
      assign(MCFile, 'LSMC.txt');
      reset(MCFile);
      readln(MCFile, NoOfQuest);
      for L := 1 to NoOfQuest do
```

and the students select option”4”, the program will turn to Fill-in the blank.

```
procedure LSFill_in_the_blanks;    {LS Fill in the blanks}
var
   NoOfQuest, K, NumOfCorrect : integer;
   FBQuestions, ModelAns : array[1..50] of string;
   FBQuest, FBAns : string;
   FBFile : text;


   procedure Read_FBFile;
   var
      K : integer;
   begin
      assign(FBFile, 'LSFB.txt');
      reset(FBFile);
      readln(FBFile, NoOfQuest);
```

The users can also choose game which same as game1 and game2.

If the users are select option"4", the scores display board will form.

```
procedure display_user_score(index : integer);
var
   percent_score : real;
begin
   if Player[index].Attempt = 0 then
      percent_score := 0
   else
      percent_score := Player[index].score/Player[index].Attempt*100;
      clrscr;
      textbackground(0);
      textcolor(4);
      writeln;
      writeln('                                Your results                    ');
      writeln('                Time: ',DateTimeToStr(now));
      writeln('                ================================================');
      writeln('                                                  ');
      writeln('                ------------------------------------------- ');
      writeln;
      writeln('                Your UserID           : ', Player[index].UserName);
      writeln;
      writeln('                Your name             : ', Player[index].EngName);
      writeln;
      writeln('                Form of Year          : ', Player[index].Form);
      writeln;
      writeln('                Total attempt score   : ', Player[index].Attempt);
      writeln;
      writeln('                Your total score      : ', Player[index].Score);
      writeln;
      writeln('                Your percentage score : ', percent_score:0:2);
      writeln('                ------------------------------------------- ');
      writeln;
      writeln('                          Press <Enter> to return. ');
      readln
end;
```

The scores display board will be display the User-name, English name, Form of year, Total attempt score, Total score and the percentage score of the user.

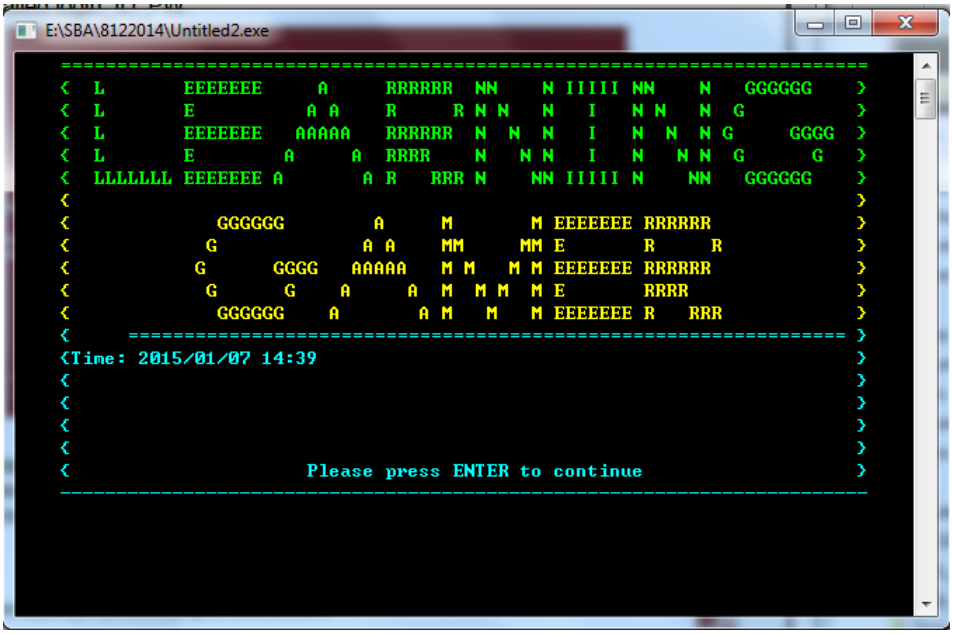
When the user select option"5", the program will call the procedure of opinion board.

```
procedure UserOpinions;
var
   opinions : text;
   UserOp   : string;
begin
   Clrscr;
   assign(opinions, 'opinions.txt');
   rewrite(opinions);
   writeln;
   writeln;
   writeln('                              Users" Opinions Region');
   writeln;
   writeln;
   writeln('          Your opinions is very important to us for improve the system!');
   writeln;
   writeln;
   write('               What is your opinions? ');
   readln(UserOp);
   writeln(opinions, UserOp);
   close(opinions);
   writeln;
   writeln;
   writeln;
   writeln;
   writeln('                         Thanks you for your opinions!');
   writeln('                            Press Enter to continue.. ');
   readln;
end;
```

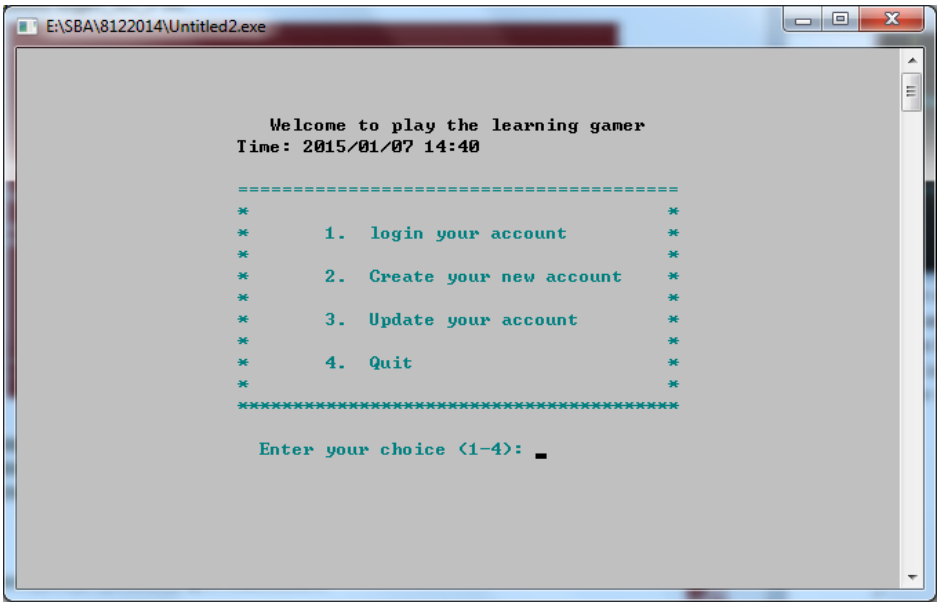
4.3) Screen layout (Execute the program)

All the test file of this program are placed at F drive that can make the program run correctly.

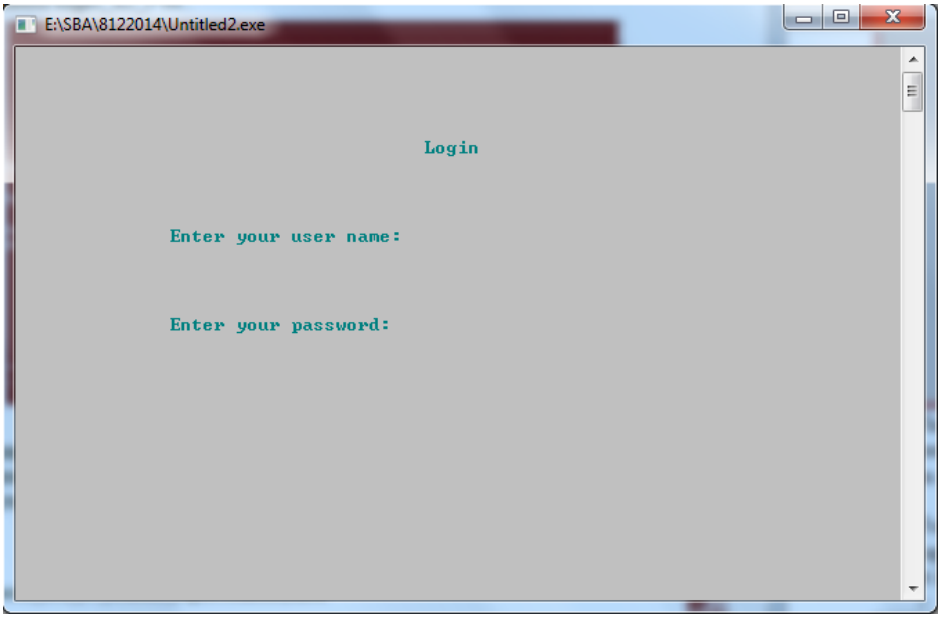
i) Welcoming Page



The above picture shows the enter welcoming page of the system. In this page, there are having a big words “Learning Gamer” and the user can click the “Enter” button for go to next page. From this enter page, It can show that the background color is black and the words in this page are formed in different color (green, yellow and light blue).

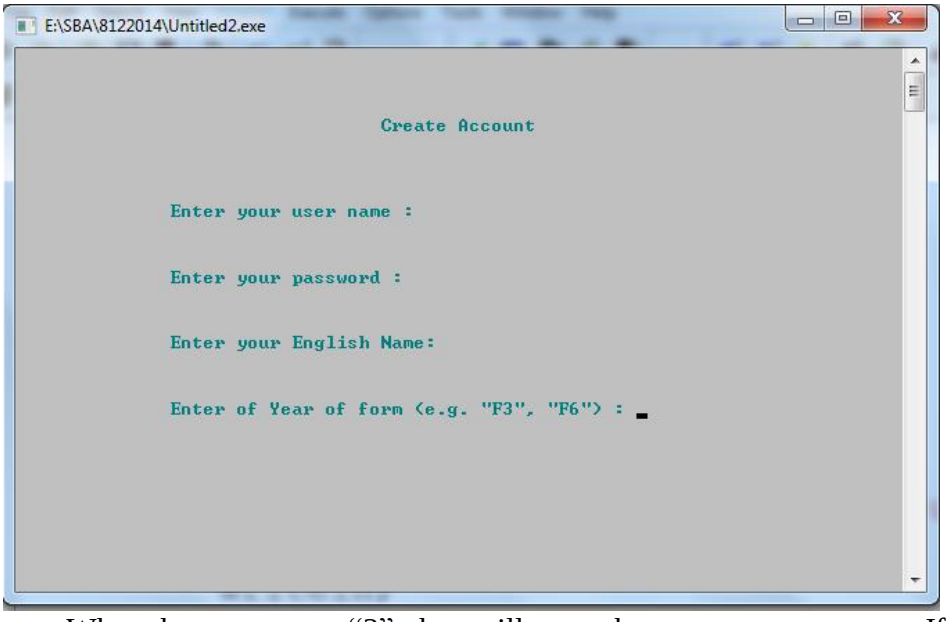


When the users press “Enter”, they can go to the login page. There are showing present time and 4 choices in the menu for the users. From this login menu, we can see that the background is white and the menu is formed by light blue.

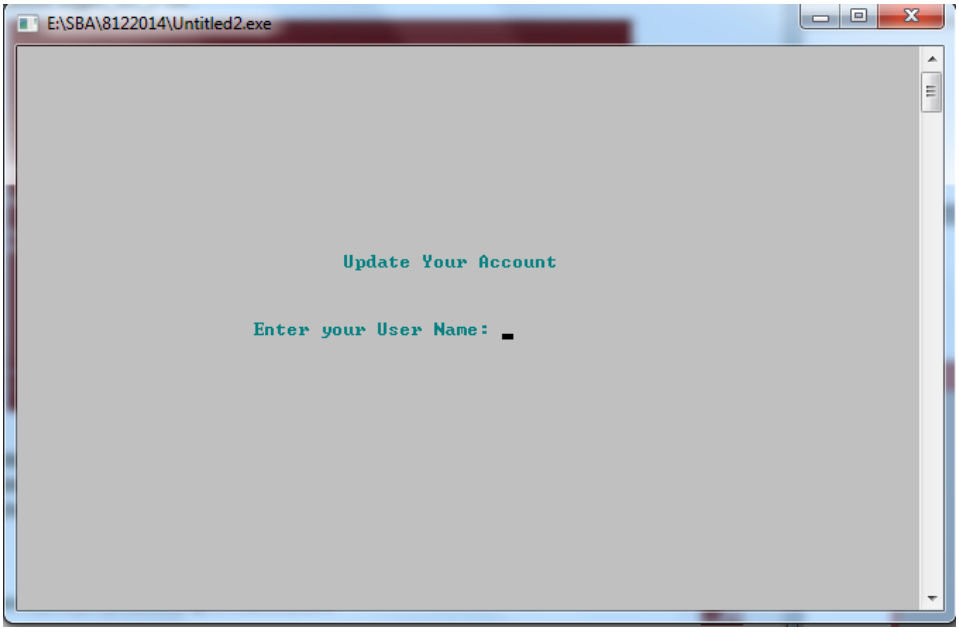


When the users press “1”, they will enter the login page. The users only need to input their valid and correct personal account, then they can enter the game system.

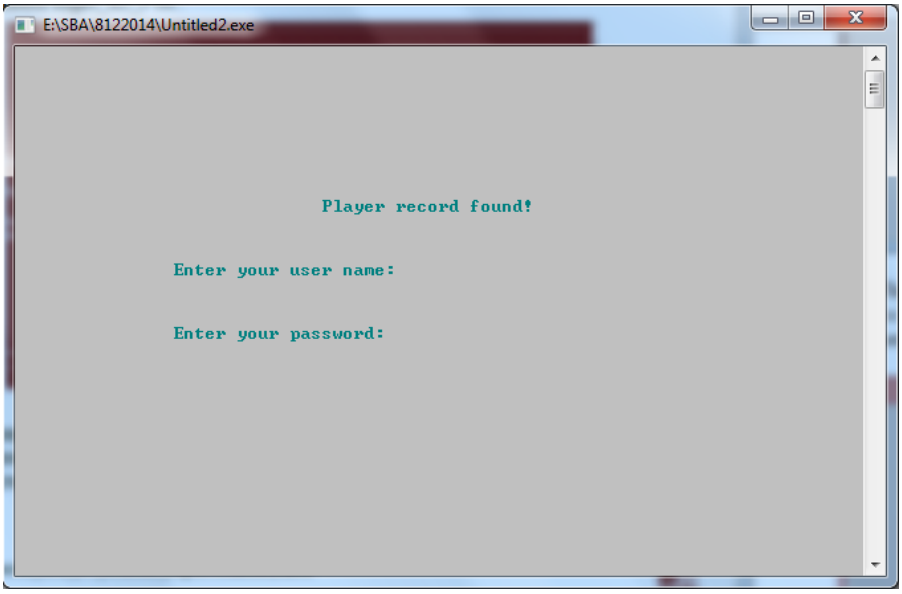
For the login, the user just only enters the correct letter. For example, if your applied user-name or password is small letter, the capital letter will not accept in this system.



When the users press “2”, they will enter the create account page. If they enter a user-name and password which is not the same with others in the text file USERAC, then the account can successfully be created.

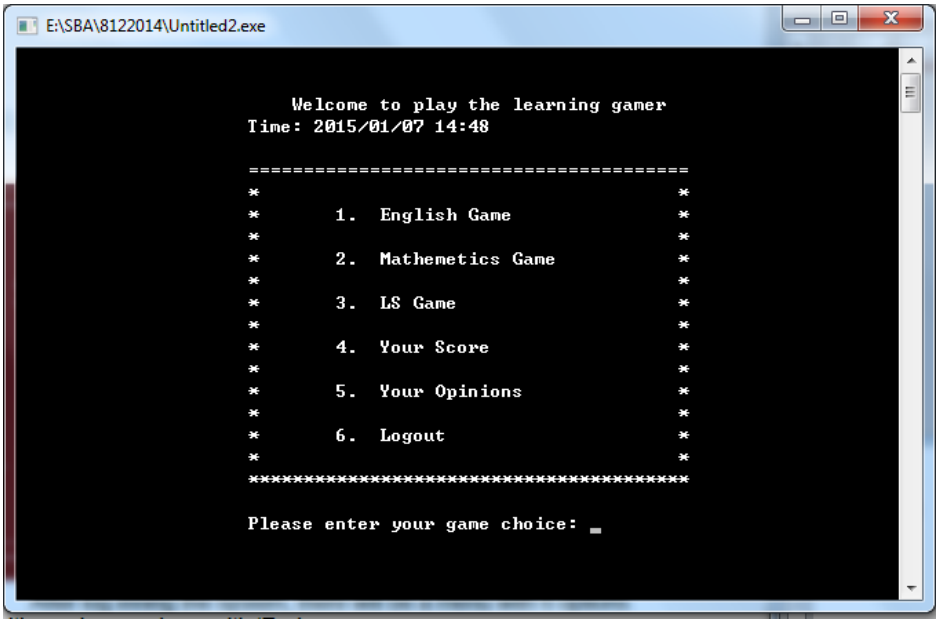


When the users press “3”, they will enter the amend account page. If they enter a user-name which has found already exist in the text file USERAC, then the user can edit their personal account.



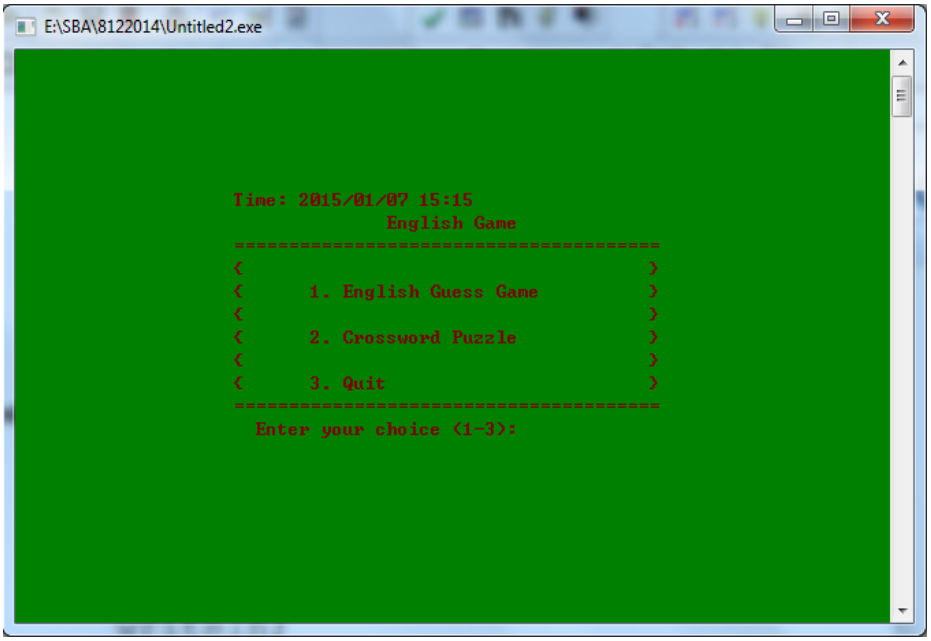
The above graph is the output that the player records are found in the text file.

ii) Main Menu

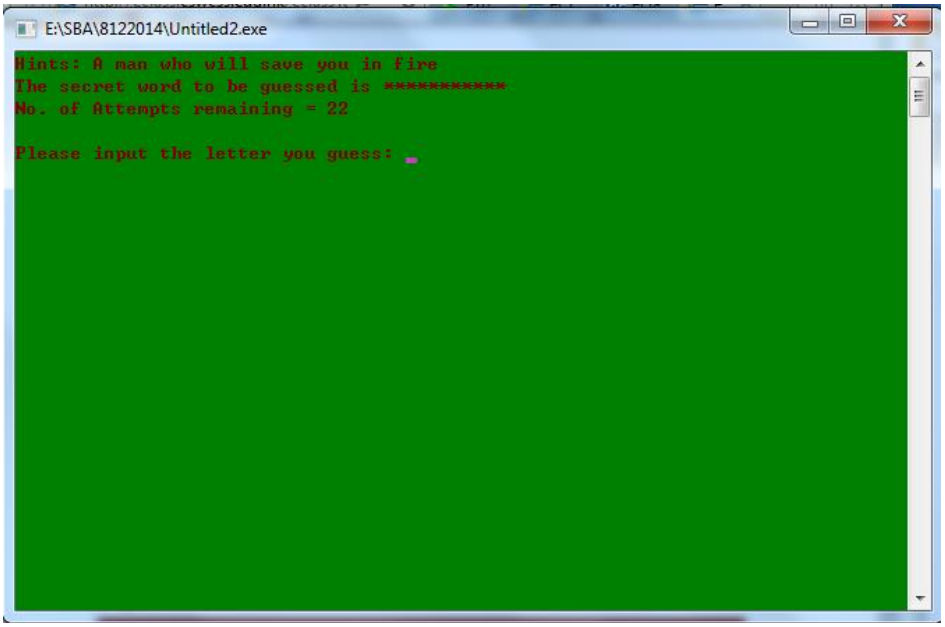


After the user login the system, the system will form the main menu with 6 options with numbers for user to choose. The layout of this page that the background color is black and the words in this page are formed in white.

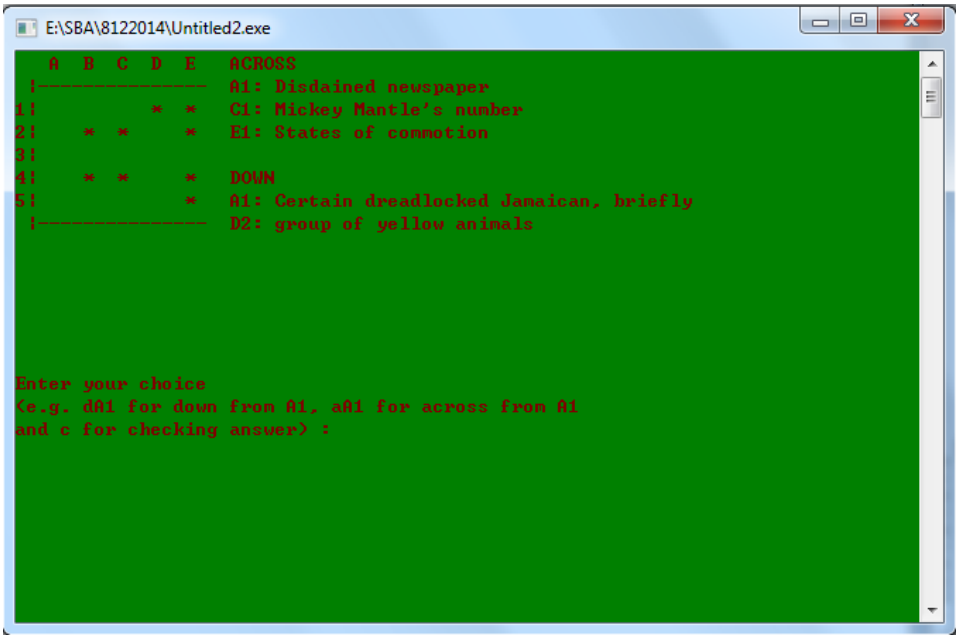
iia) English Game



When the users press "1" in the main menu, it will turn to the English game page. In this page, the layout of this page background is formed by green and the word color is formed by red.

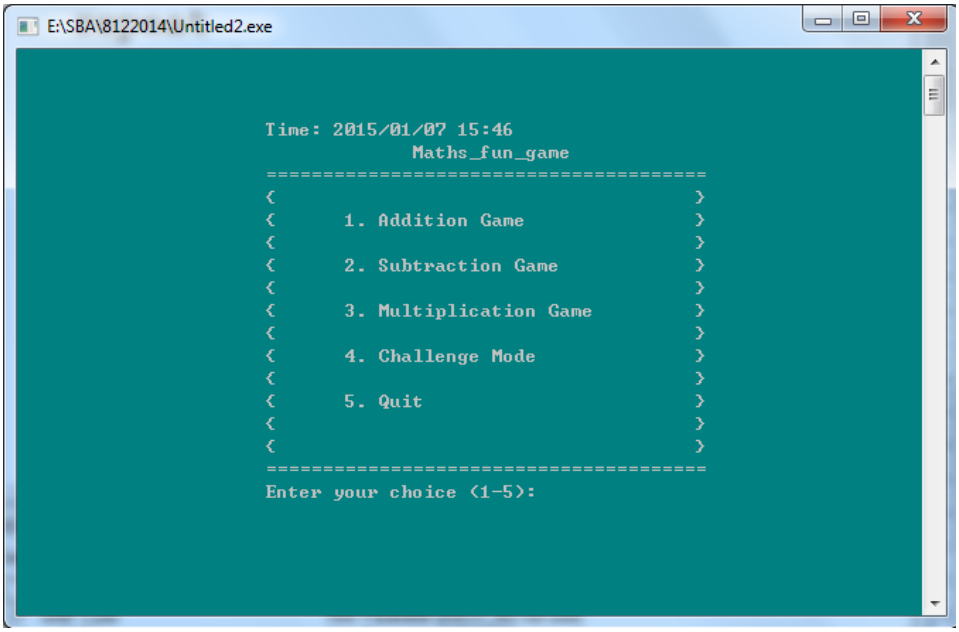


When the users press "1" again, it will turn to the Word-guess game. This game needs to guess the word with the hints and after finish the question, the meaning of the word with form.

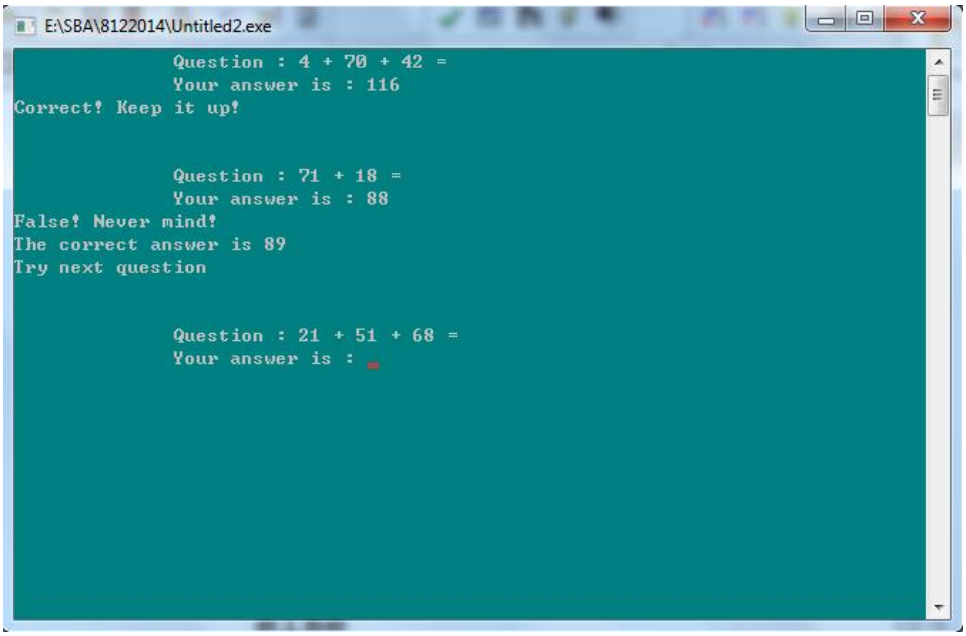


When the users press "1" again, it will turn to the Crossword game. This game as same as traditional Crossword puzzle but the table are formed in 5x5 tables. The user just follows the hints beside the table and enters the words.

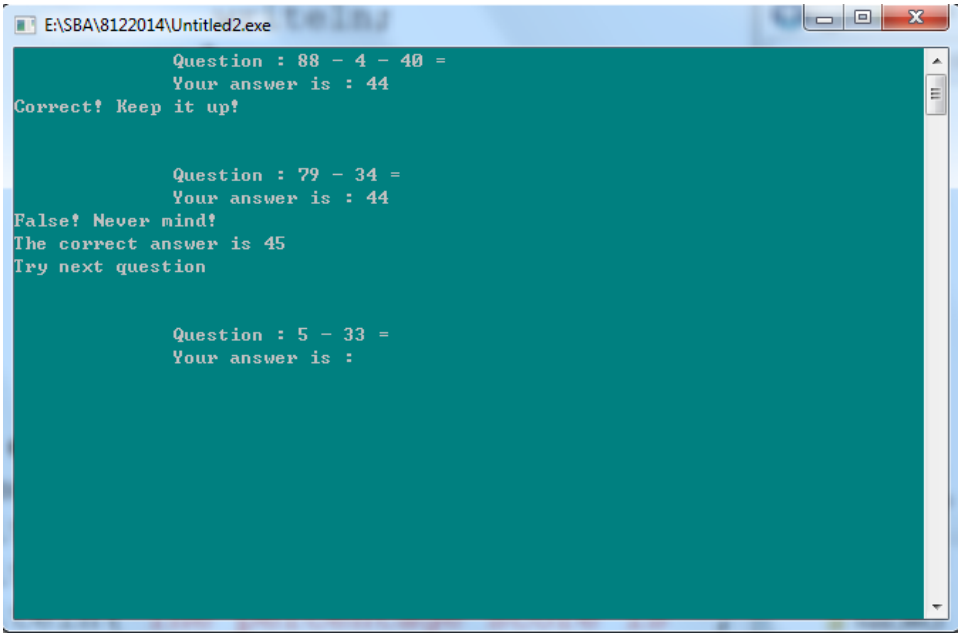
iib) Mathematics Game



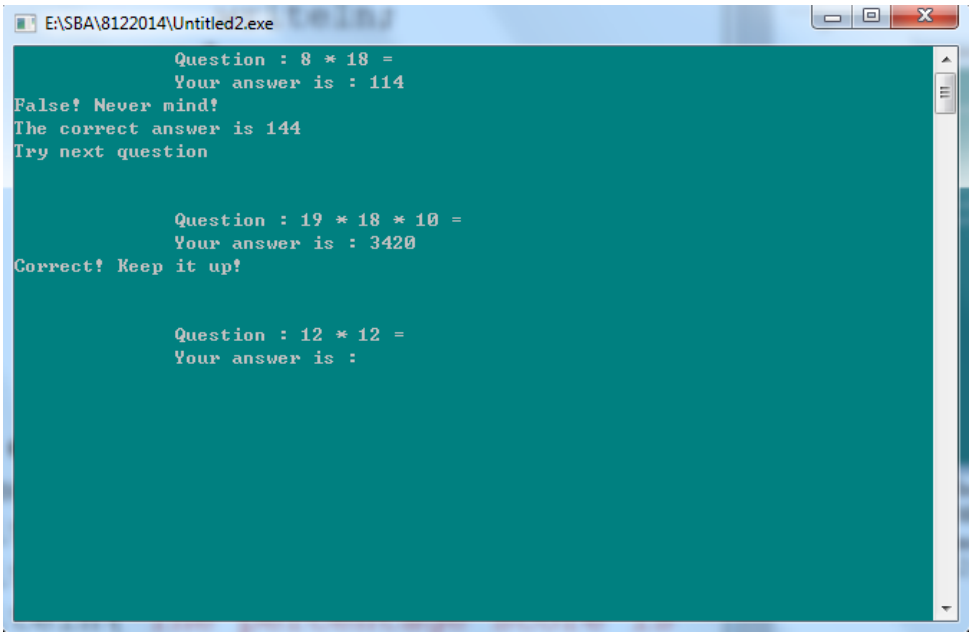
When the users press "2" in the main menu, it will turn to the Mathematics game. In the Mathematics game, the background color of the layout is baby blue and the words color are white.



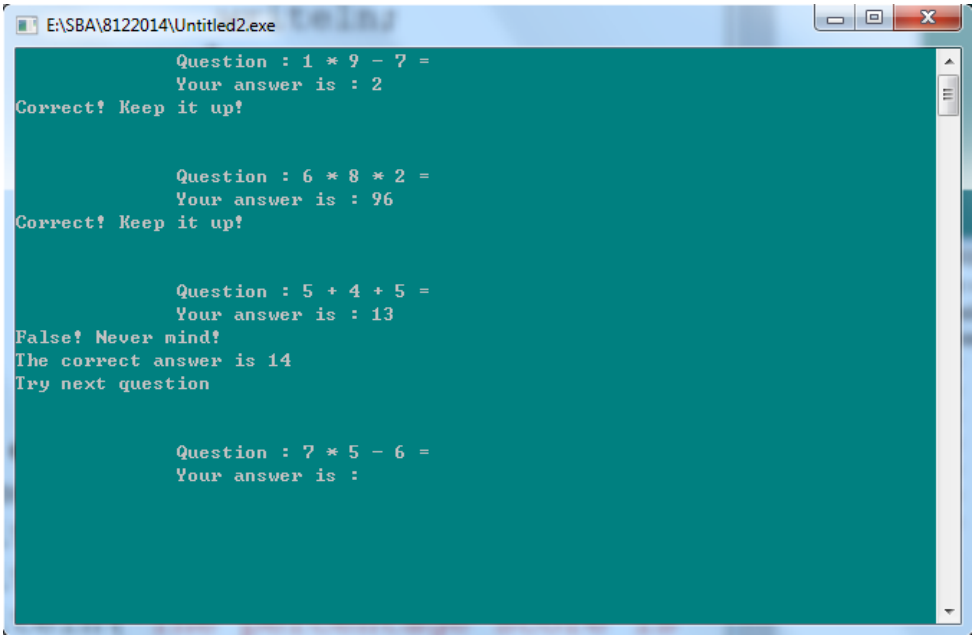
When the users press "1", the system will turn to the addition game of the Mathematics game. The user need to answer the 10 questions of addition way and each questions are generate by the computer randomly with 2 parts or 3 parts.



When the users press "2", the system will turn to the subtraction game of the Mathematics game. The user need to answer the 10 questions of subtraction way and each questions are generate by the computer randomly with 2 parts or 3 parts.

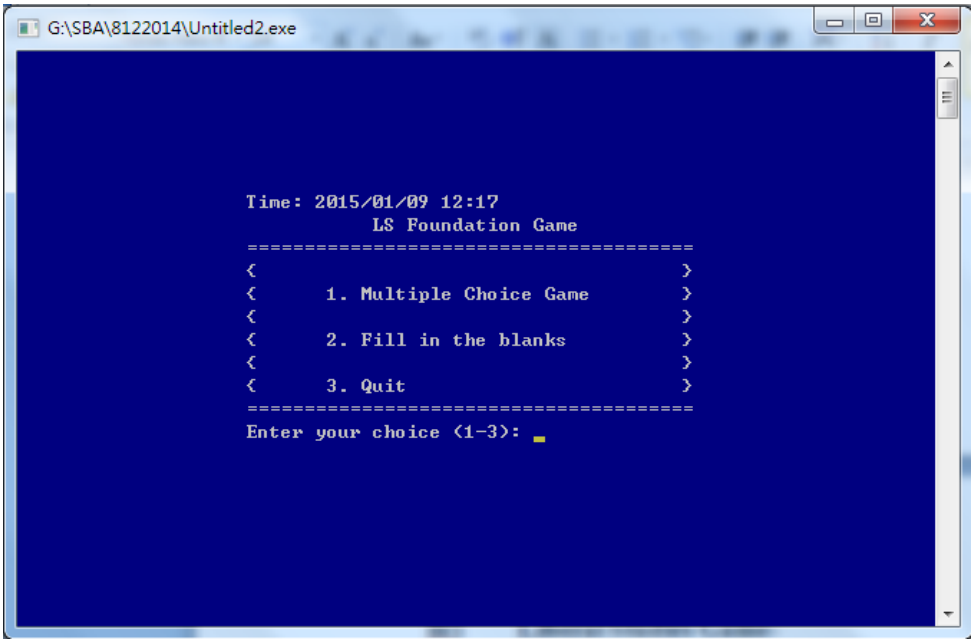


When the users press "3", the system will turn to the multiplication game of the Mathematics game. The user need to answer the 10 questions of multiplication way and each questions are generate by the computer randomly with 2 parts or 3 parts.

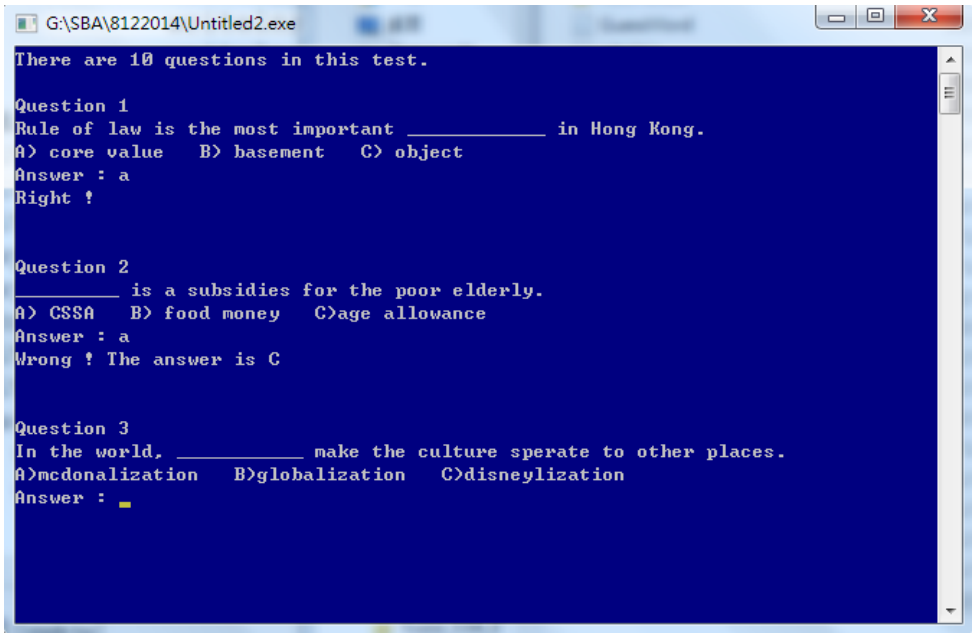


When the users press “4”, the system will turn to the challenge mode of the Mathematics game. The user need to answer the 10 questions of mixture way with addition, subtraction or multiplication, the user need to follow the rule of arithmetic to answer the question and each questions are generate by the computer randomly with 2 parts or 3 parts.

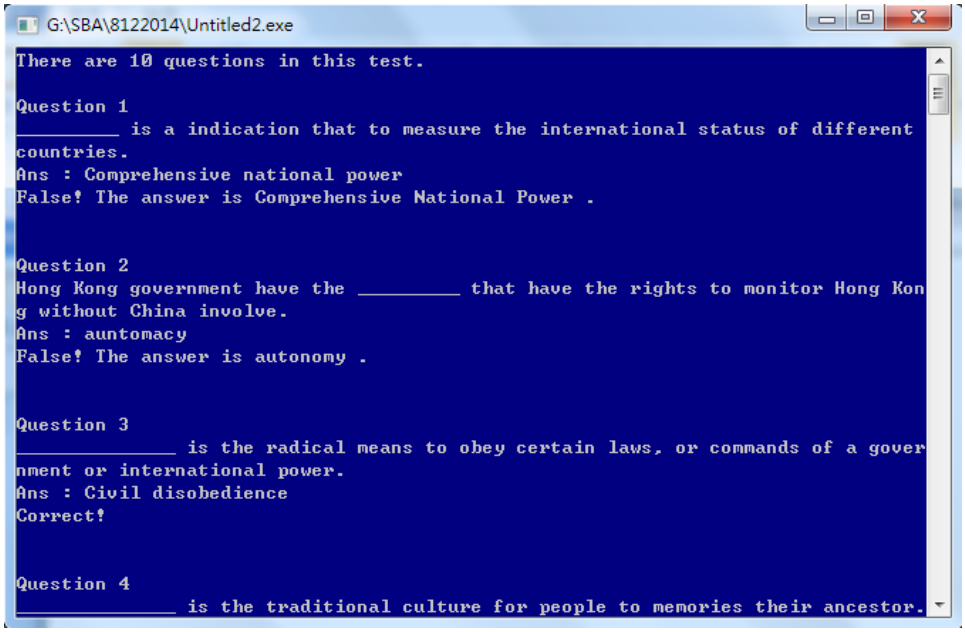
iic) Liberal Studies Game



When the users press "3" in the menu, it will turn to the liberal Studies game. In the Mathematics game, the background color of the layout is blue and the words color are white.

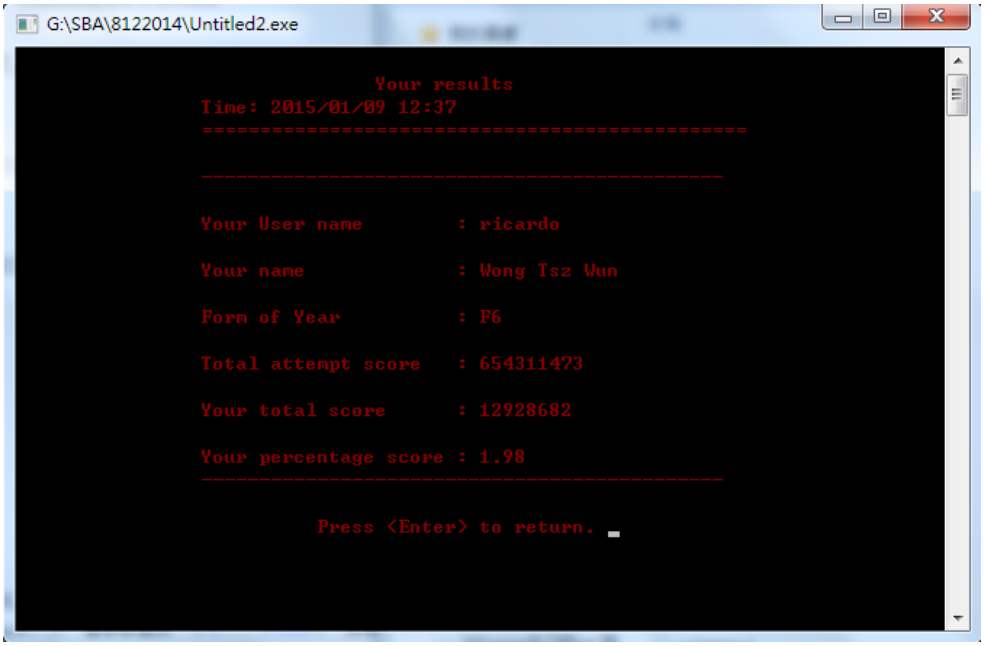


When the users press "1", the system will turn to the Multiple choice game of the Liberal Studies game. There have 10 questions and user needs to choose 3 answers which in A to C. The user can enter their choice in both capital or small letter.



When the users press “2”, the system will turn to the Fill in the blanks game of the Liberal Studies game. There have 10 questions and user needs to fill the blanks with some terms and the user need to follow the capital or small letter of the sentence when they answer the question.

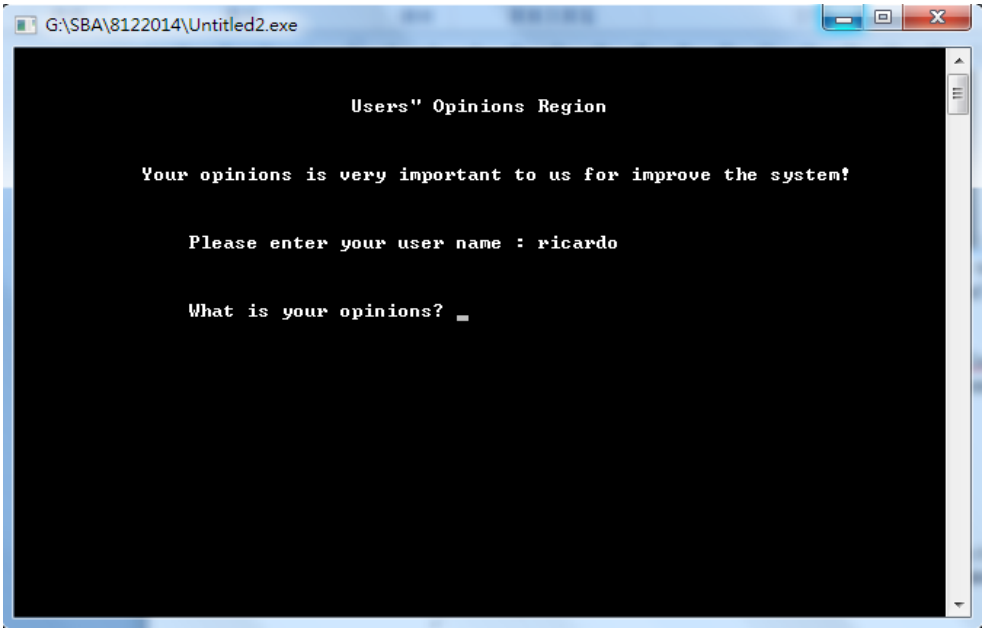
iid) Scores display



When the user press ”6” in the main menu, the system will turn to the scores display board. From the above graph, the background color of this layout is black and the words colors are red.

In this page, this table in this page will be show the present time, UserID, English name, form of year, total attempt score, total score and the percentage score.

iie) Opinion-Region



When the user enters ”6” in the main menu, the system will turn to the opinion region. In this page, the background color is using black color and the

words color is using white.

In this region, the user need to enter their user name and their opinions and all of their input information will be store into a opinion text file.

5) Testing & Evaluation

5.1) Description

In this program, there will be having some loopholes and bugs, those bugs will contain logical error or run-time error. In will affect the quality of the system. So, this part is used to check whether the program can work probably and achieve the goals of this task.

First of all, I have to build up the simple structure of the program which is from the design. I have to analyze all of the functions and set up test cases to find out whether the program can handle all the situations that may cause errors.

Secondly, the testing will be involve the participation of the user as there will having some bugs hiding in the program which cannot find in the debugging process. So the testing part is still continue when the users are playing.

Thirdly, as the reason why I found out the bugs and loopholes is that I have to do some debug works to make the quality of the program become well, so debugging will be followed by testing.

In this section, I will do the testing of the program and do the evaluation of the program that to make sure the program can have the well performance for the users play it.

5.2) The planning of the Testing and Evaluation

The program will be tested and evaluated under the plan below.

When the program is on half of its way, I will test the program first. It is important for me to test it in first time as there may have the errors that users can't discover easily. For each of the games and function, at least three times testing should have been done. There will be a lot of possibilities for the bugs, so both correct and incorrect data should be tried to ensure the accuracy of the program. After finding all the bugs, debug is required and at the same time, I also need to modify the program as some difficulties will be find out while testing the program.

When the program is completed, I will do the second time testing and debugging works that this time is going to find out the defect of the program and to improve the quality of the program.

But still, there are some of the problems cannot find as there will involve some subjective on the works. So, in the final testing, I will invite other students to play my program and see whether there are any bugs exist.

5.3) Testing and Debugging

There could be three types of error are often occurs in the program, which is syntax, run-time and logic error.

1. **Syntax**:

   In this program, there are always have some syntax errors. Some of them are typing mistakes which is a transposition error. Some of them are missing begin or end statement; some of them are missing semicolon; some of them missing the period. Also sometimes, I define an incorrect type of variable to it. For example, word should be defined as string but the variable is character, so the error will be occur. It is quite easy for me to debug the above syntax errors, since Dev-Pascal provide a strong debugger which enables the user to locate the errors. As a result, I can correct all errors in the program by using Dev-Pascal' debugger.

2. **Run-time:**

   The run-time error always is quite brothering me the run the program. Moreover, it is harder to debug as Pascal do not warn the programmer where is the error. However, once the programmer executes the program, it will automatically lose it due to run-time error. After debugging, I found out that usually they occur because of incorrect route of the file, and enter a word which not matches with the type of the variable. Furthermore, it will also occur when the I compile a file and write something to it, and I forgot to close it. As a result, when I do the action again, run-time error occurs.

3. **Logic:**

   It is the hardest part of error for me to debug. Since neither Pascal nor the program will show errors to you. The logic error will hard to find out where are the errors. Only if enter the testing data can find the errors. For example, check the range of the question or check the answer from the user. If we cannot have a correct logic, the program will be error. So it will be the hardest error of the program.

Here are several testing ways are carried out to ensure the program is workable

1. **Unit Test**:

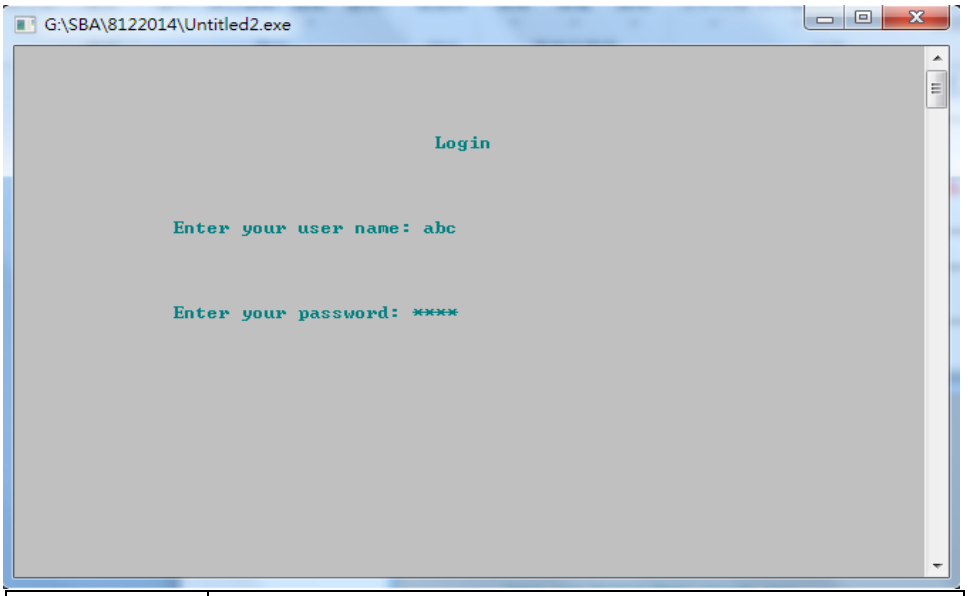
   I have tested the different individual modules before they are integrated into a big program with other procedures. The purpose of the way is to identify and fix as many errors as possible before they are jointed in to a larger program. It can decrease the workload when I am debugging.

2. **System Test:**

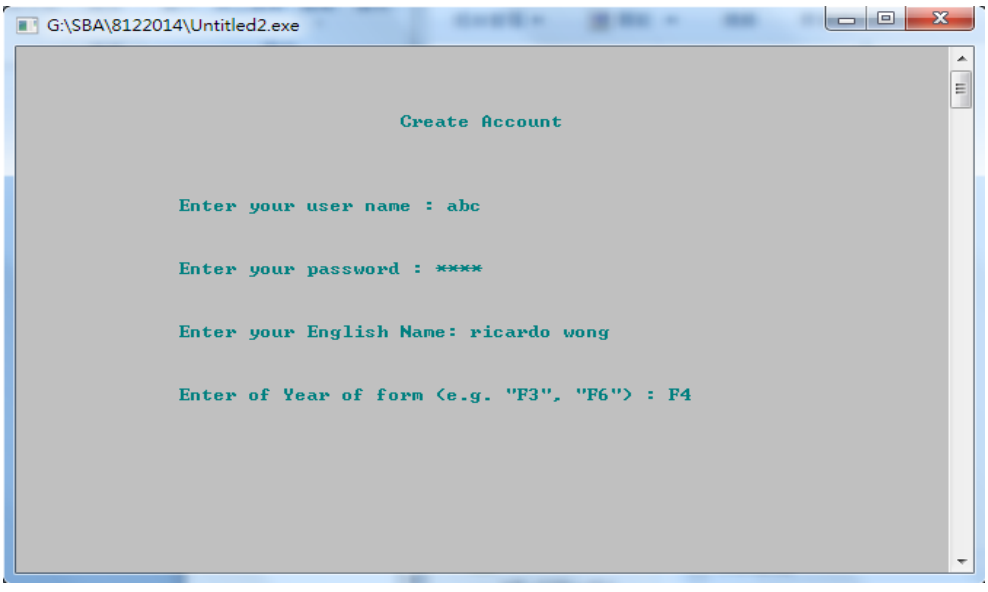
   It is conducted after the unit test has completed. It does not test individual modules while it tests the larger and completed program. Its goal is to test and compare with the user and the questions' requirement. Moreover, it enables me to locate the error between different procedures. If there are any procedures are not workable with other procedures, I can fix it much easier.

5.2) Testing code

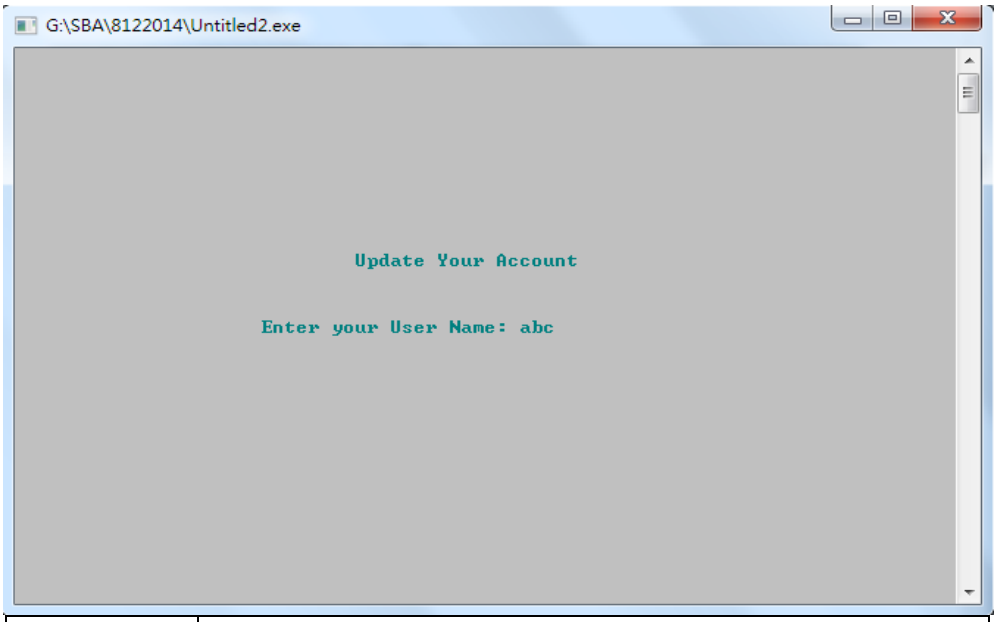
Here are the testing records after the final debug for the program:



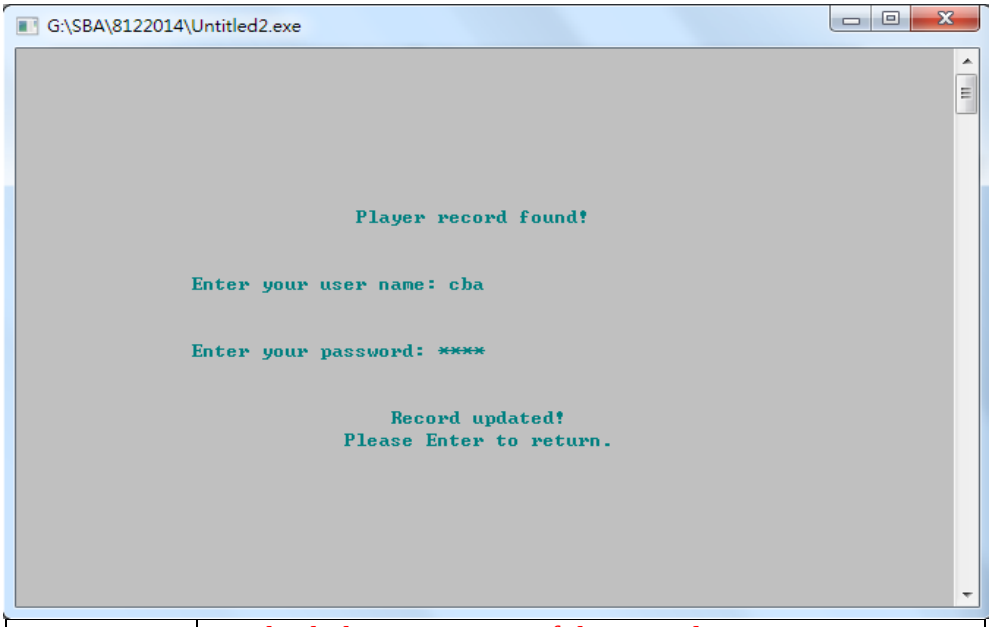
Purpose:	To check the correctness of the input of the login system.
Input:	User-name, Password (another valid account)
Expected Output:	Welcome page appear
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found.
Follow-up Action:	Nil



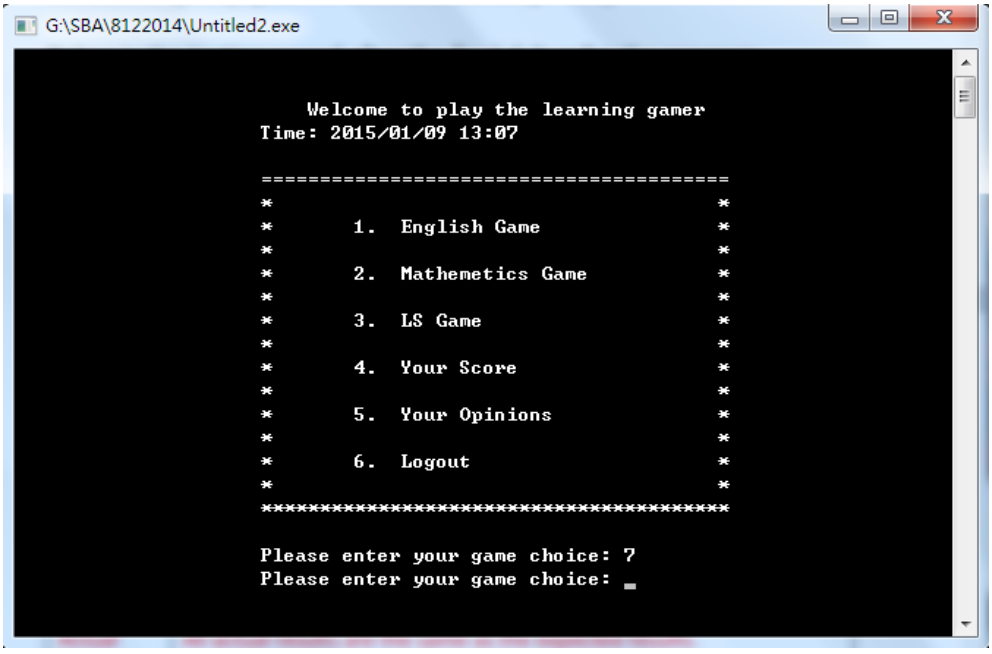
Purpose:	To check the correctness of the create account system
Input:	User-name, Password, English name, Form of year (another valid input)
Expected Output:	Back to the login menu
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found.
Follow-up Action:	Nil



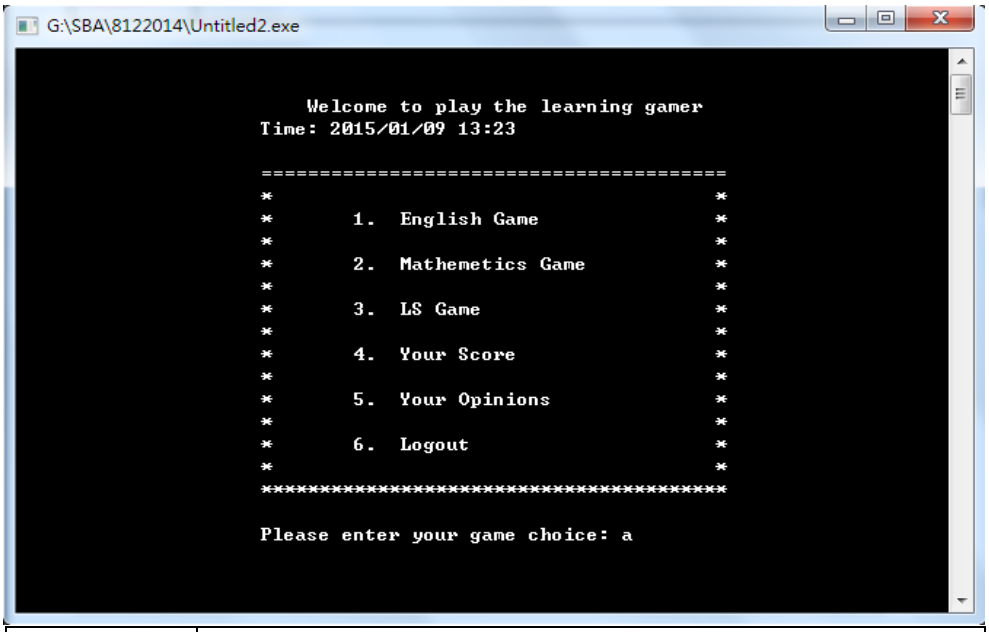
Purpose:	To check the correctness account searching of the amend account system
Input:	User-name (another valid User-name)
Expected Output:	User account found
Actual Output:	User account cannot found
Test Result:	Bugs found. Going to the wrong procedure.
Follow-up Action:	Nil



Purpose:	To check the correctness of the amend account system
Input:	User-name, Password (another valid account)
Expected Output:	String “Record Updated” appear
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found.
Follow-up Action:	Nil



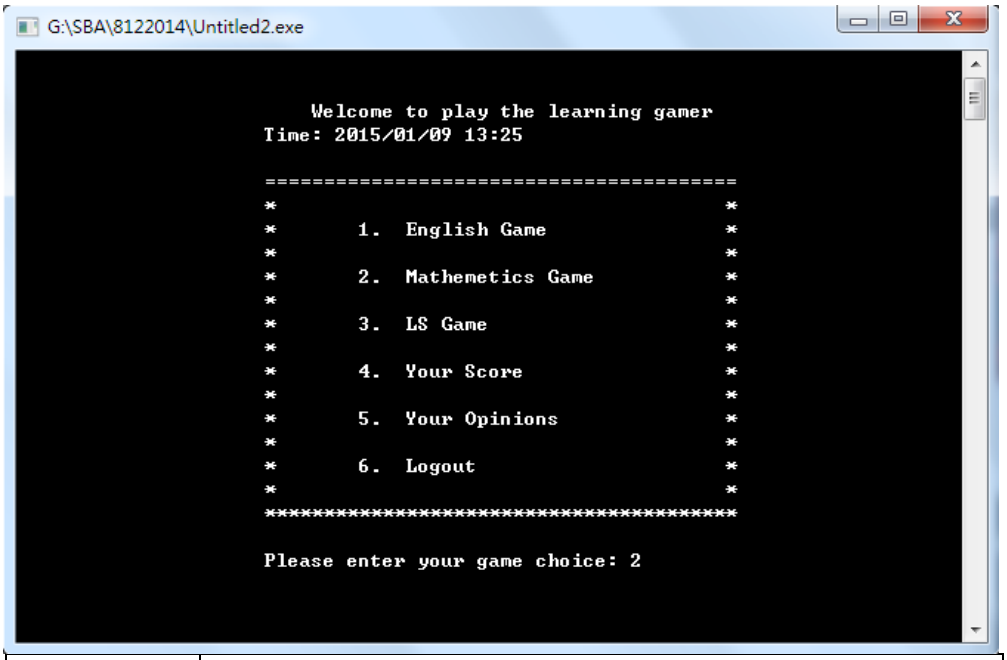
Purpose:	To check if the program can detect the invalid input
Input:	Choice which is out of range
Expected Output:	The program will repeat the “Please enter your game choice: “statement.
Actual Output:	All actual results are the same as the expected results
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



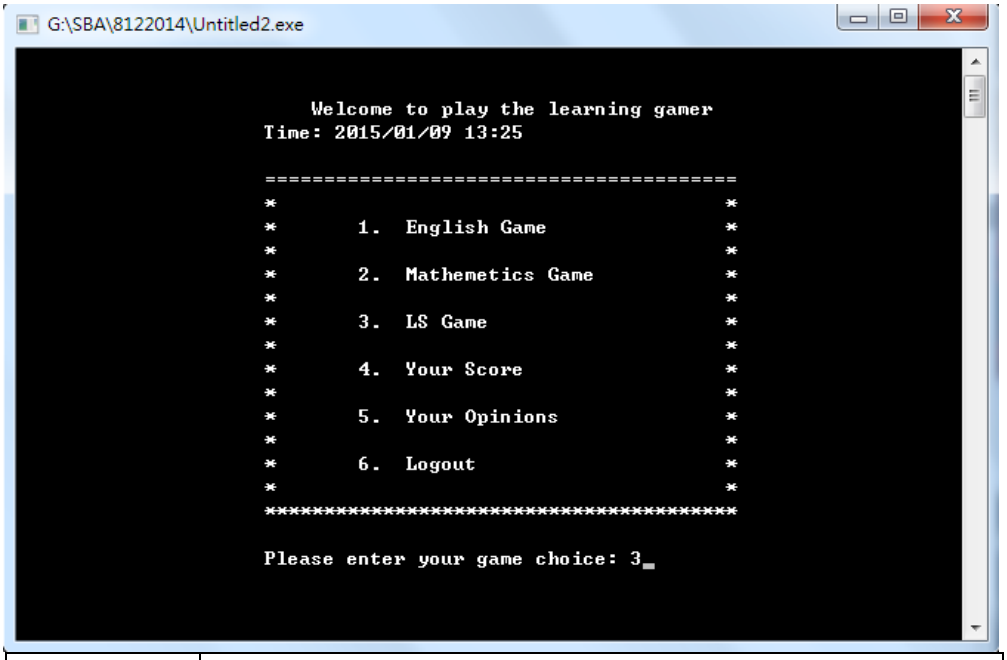
Purpose:	To check if the program can detect the invalid input
Input:	Input another type of data
Expected Output:	The program will repeat the “Please enter your game choice: “statement.
Actual Output:	The program closed.
Test Result:	Bugs found.
Follow-up Action:	Nil

Purpose:	To check the correctness of the output results on screen by using invalid data.
Input:	* (another invalid data)
Expected Output:	The program will not have any output or changes
Actual Output:	All actual results are the same as the expected results
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

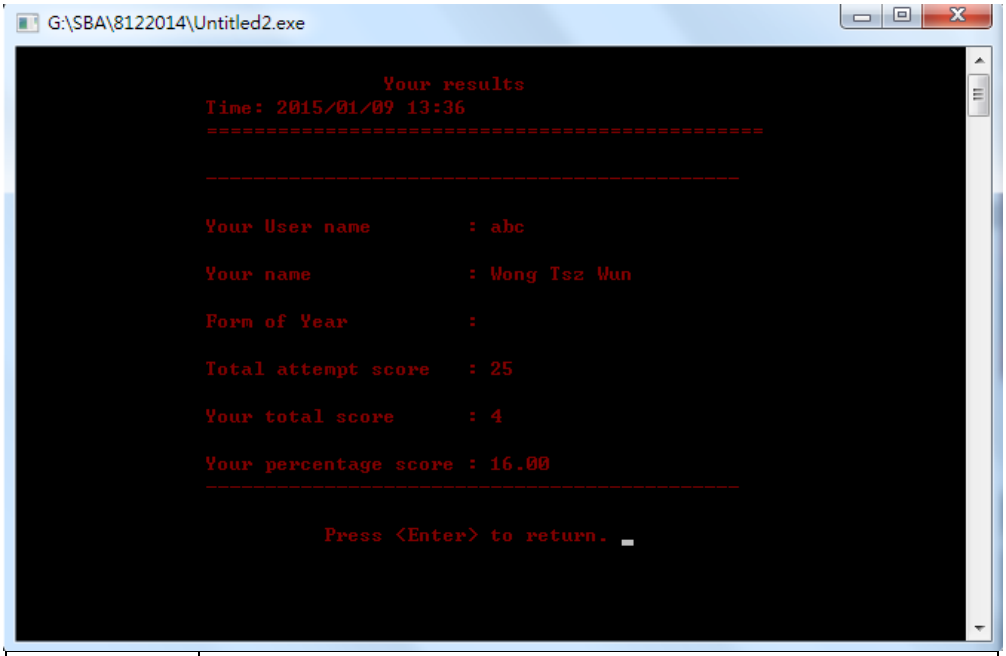
Purpose:	To check if the program can jump to the specific procedure when entering a correct input.
Input:	1 (one of the choice)
Expected Output:	Jumping into English game
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil



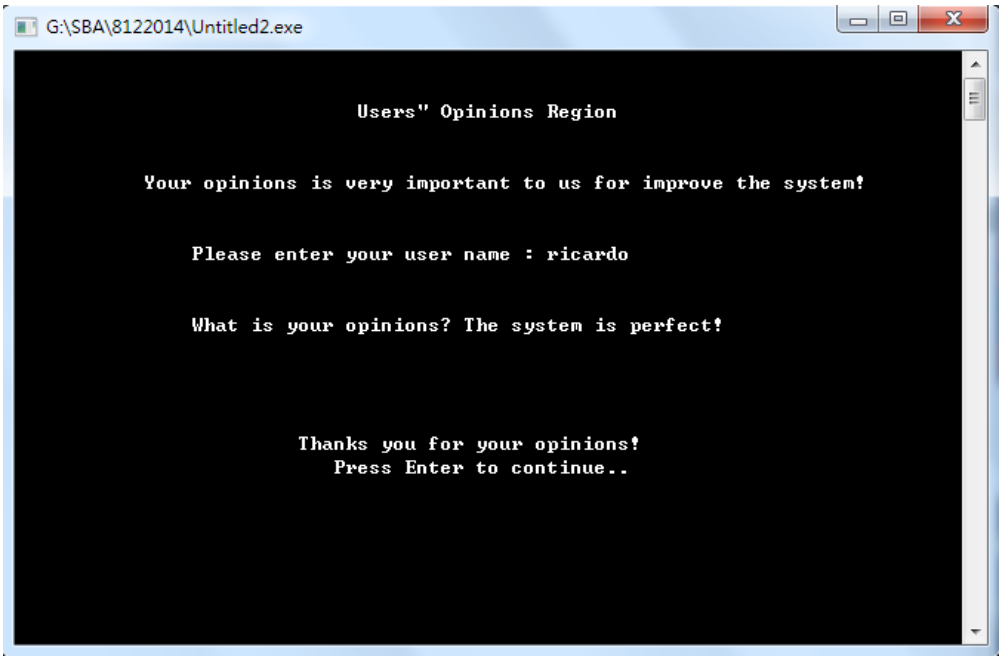
Purpose:	To check if it can go to the specific procedure with another data.
Input:	2 (another data different from the above test)
Expected Output:	Going to Mathematic game
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found.
Follow-up Action:	Nil



Purpose:	To check the correctness of the output results on screen by using another input data.
Input:	3 (another choice)
Expected Output:	Jumping to Liberal Studies procedure.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found.
Follow-up Action:	Nil



Purpose:	To check the display board of the system
Input:	Nil
Expected Output:	All the information and the scores formed
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found.
Follow-up Action:	Nil



Purpose:	To check the correctness of opinion region of the system
Input:	User-name and the opinion of the user
Expected Output:	String “Thanks you for your opinions!” appear
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found.
Follow-up Action:	Nil

5.4) Further improvement (summary of the forms)

1. A different version should be provides to different type of users. As the above, this main user of this program is the teachers and students. The different versions of the program can make it more user-friendly for different user. For example, for teachers, the program can provides those teachers to edition the question for the students. So is can make the program more comprehensive..

2. In this system, there is not having the ranking of the students. So, for the improvement, the ranking system should be approach. For example, the system can find 10 students who scores is having top 10. It can make the students to know about their ranking in this program.

3. We can make the program become more interesting. In this program, there is not having too interesting games in there. It may not attract the students to play it. It would be much more interesting, if there is any graph or movement in the game. So that is can make the students more

4. There is not having the further system to update the program sources continuously. It will make the program cannot update in one period of time. So I think the program should have the update system to make the program have the improvement continuously. Such as update the vocabulary in guess word game or the question in LS game. If there can have the update system, it will increase the interesting and comprehensive of the program and

6) Conclusion and Reflection

6.1) Conclusion

I believe using this program can mostly fulfill the requirement for providing an information system for the junior form students or post-secondary students to play English vocabulary games which improve their knowledge of the core subject. By playing different games in the program, I am sure it can improve the knowledge of the different games of the different subject of the student. It is because by testing the program, I also learn some knowledge form it. As a result, I do think it is quite useful for them to play this program.

I also found out that interesting is the key to success. As when I test the program at the first, I felt it was very boring. Therefore, I add some functions and tricks on it so that it would be funnier. If you are interested, you would learn faster than anyone else.

I think my program is good for enriching the useful knowledge of different subject since the games are generated by different catalogue. Therefore, the students can learn in different aspects but not only a specific perspective. Moreover, my program provides a greater range of learning subjects. If the students just know the specific subjects, it would be useless. Only if the students learn a proper knowledge, can he/she improve his/her proficiency. Besides, my program can show the percentage of correctness of each game, so that the students can aim at a higher or even full mark in each game, while their teachers can know exactly how well or bad is their students are and make them to know about their goals in future..

But all in all, the program can work properly and smoothly which can make the users to use it with a higher efficiency. I am sure that I can do better next time after this program as I can have a good experience of making a program and my time arrangement skills will be better. I will improve the program with more comprehensive tasks.

To conclude, I think this program is suitable for the students to learn the knowledge of different subject in schools.

6.2) Reflection

After doing this project, I found out that it is quite important for one person to think critically. As a result, it enables you to find out different solutions for a problem. Sometimes, when I try to debug a logic error, I have to use dry-run table which I learnt from the lesson. Also, I learn several commands for my program too. For example, clrscr, uses, and so on. The above commands cannot find in the textbook. However, it is much convenient than using in the program. Besides, this project lets me know that when doing a big project, it is important for us to use different procedures and functions. This can not only reduce the length in main program, but also easier for us to debug if there is any error occurs.

Before I do this, my knowledge about this topic are very limited and don't know many things. Since I do this topic, I read much information on the Internet, the textbook and ask the supervising teacher's to increase my related concept and knowledge. When I do this study, it can make me have a clear concept of this topic and my extra knowledge of the Pascal Programming. So, I can also prevent many digression appear on this study and make me do this study without much confusion and do it clearly.

I can also learn about organizing and collecting of data. It was difficult to build up this program before I start, and I think it is very difficult to do as there are many unknown knowledge came out when I found the information, notes from teacher and textbook. But I think it is very interesting experience for me. Moreover this study can help me to develop the programming skills and encourage the appetite for knowledge. I think I also can though this opportunity to encourage me to do more programming work and can help me to learn to do the similar study next time. So I think this experience is good for me to learn in the ICT.

7) Appendix

7.1) Coding

```
program GameEdition;

uses crt, sysutils, dos, windows;

const
  max_user = 1000;

type
   UserACType = record
                 UserName : string[30];
                 Password : string[50];
                 EngName  : string[50];
                 Form     : string[5];
                 Score    : integer;
                 Attempt  : integer;
                end;

var
  index, numuser : integer;
  Player : array [1..100] of UserACType;


procedure ReadRecord;
var
  count : integer;
  UserAccount : text;
begin
  assign(UserAccount, 'userac.txt');
  reset(UserAccount);
  count := 0;
  while not eof(UserAccount) do
  begin
    count := count + 1;
    with Player[count] do
      begin
        readln(UserAccount, UserName);
        readln(UserAccount, Password);
        readln(UserAccount, EngName);
        readln(UserAccount, Form);
        readln(UserAccount, Score);
        readln(UserAccount, Attempt);
      end
```

```
   end;
   numuser := count;
   close(UserAccount)
end;


procedure store_users_info;
var
   i : integer;
   users : text;
begin
   assign(users, 'userac.txt');
   rewrite(users);
   for i := 1 to numuser do
      with Player[i] do
         begin
            writeln(users, UserName);
            writeln(users, Password);
            writeln(users, EngName);
            writeln(users, Form);
            writeln(users, Score);
            writeln(users, Attempt);
         end;
   close(users)
end;

{***********English Game************}
procedure EnglishGame;
var
   Selection : integer;
   numQ, numCor : integer;

   procedure EngGuessGame;        {Start of Guessword game}
   var
      Secret, Solution, Hint_Strg, Meaning : string;
      Done : boolean;
      Letter : char;
      NoOfAttempts : integer;
      word : array[1..50] of string;
      hint : array[1..50] of string;
      mean : array[1..50] of string;
      NoOfWord : integer;
      Continue : char;

      procedure readwords;
```

```
var
   i : integer;
   infile : text;
begin
   assign(infile, 'GuessWord.txt');
   reset(infile);
   i := 0;
   while not eof(infile) do
    begin
       i := i + 1;
       readln(infile, hint[i]);
       readln(infile, word[i]);
        readln(infile, mean[i]);
    end;
   NoOfWord := i;
   close(infile)
end;

procedure SelectSecret(var Secret, Hint_Strg, Meaning : string);
var
   m : integer;

begin
   randomize;
   m := random(NoOfWord)+1;
   Secret := word[m];
   Hint_Strg := hint[m];
   Meaning := mean[m]
end;

procedure Init;
var
   i : integer;

begin
   Solution := '';
   for i := 1 to length(Secret) do
    Solution := Solution + '*';
     NoOfAttempts := 2 * length(Secret)
end;

procedure LetterGuess;
var
   i : integer;
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)
  end;

  procedure WinCheck;
  begin
    Done := (Solution = Secret) or (NoOfAttempts = 0)
  end;

begin       {Main body of English Guess Game}
  numCor := 0;
  numQ := 0;
  readwords;
  repeat
    clrscr;
    numQ := numQ + 1;
    SelectSecret(Secret, Hint_Strg, Meaning);
    Init;
    Done := false;
    writeln(Hint_Strg);
    writeln('The secret word to be guessed is ',Solution);
    writeln('No. of Attempts remaining = ',NoOfAttempts) ;
    writeln;
    repeat
      LetterGuess;
      WinCheck
    until Done = true;
      if Solution = Secret then
        begin
          numCor := numCor + 1;
          writeln('You win!!');
        end
      else begin
            writeln('You lose!!');
            writeln('The Answer is ', Secret);
          end;
    writeln(Meaning);
    Player[index].Score := Player[index].Score + 5*numCor;
    Player[index].Attempt := Player[index].Attempt + 5*numQ;
```

```
      store_users_info;
      writeln;
      write('Try again(Y/y) to continue? ');
      readln(continue)
   until (continue <> 'Y') and (continue <> 'y');

end;   {End of English Guess Game}


procedure CrossWPuzzle;       {Start of crossword puzzle}
const  max_no_q = 4;
var
   hint_num, quest_num, num_corr, num_char : integer;
   word, content : array[1..5, 1..5] of char;
   hint: array[1..25] of string;
   choice : string;
   valid_input, stop_game : boolean;
   try_again : char;
   quest_no_str : char;


   procedure Get_Words(quest_no : integer);
   var
      wordfile : text;
      i, j : integer;
   begin
      assign(wordfile, 'WORDS' + quest_no_str + '.txt');
      reset(wordfile);
      num_char := 0;
      for i := 1 to 5 do
         begin
            for j := 1 to 5 do
              begin
               read(wordfile, word[i,j]);
              if word[i,j] = ' ' then
                 begin
                    word[i,j] := '*';
                    content[i,j] := '*'
                 end
               else
                 begin
                    content[i,j] := ' ';
                    num_char := num_char + 1
                 end
             end;
```

```
        readln(wordfile)
      end;
  close(wordfile)
end;


procedure Get_Hints(quest_no : integer);
var
  hintfile : text;
  i : integer;
begin
  assign(hintfile, 'HINTS' + quest_no_str + '.txt');
  reset(hintfile);
  i := 0;
  while not eof(hintfile) do
    begin
      i := i + 1;
      readln(hintfile, hint[i])
    end;
  hint_num := i;
  close(hintfile)
end;

procedure Display_Table(Row, Column :integer);
var
  i, j : integer;
begin
  write(' ');
  for j := 1 to Column do
     write(chr(64+j):3);
  writeln;
  write(' |');
  for j := 1 to Column do
     write('---');
  writeln;
  for i := 1 to Row do
     begin
       write(i);
       write('|');
       for j := 1 to Column do
         write('   ');
       writeln;
     end;
  write(' |');
  for j := 1 to Column do
```

```
      write('---');
   writeln;
end;

procedure Display_Content(Row, Column :integer);
var
   i, j : integer;
begin
   for i := 1 to Row do
      for j := 1 to Column do
         begin
           gotoXY(3*j+1,i+2);
             write(content[i,j])
         end;
   gotoXY(1,2*Row+4);
end;

procedure Display_Answer(Row, Column :integer);
var
   i, j : integer;
begin
   for i := 1 to Row do
      for j := 1 to Column do
         begin
               gotoXY(3*j+1,i+2);
           write(word[i,j])
         end;
   gotoXY(1,2*Row+4);
end;

procedure Display_Hints;
var
   i : integer;
begin
   for i := 1 to hint_num do
      begin
        gotoXY(20,i);
        writeln(hint[i])
      end
end;

procedure Get_Input;
var
   wd : string;
   C, R, P, i, j : integer;
```

```
begin
   for i := 15 to 20 do
      begin
         gotoXY(1,i);
         writeln('                                                                    ');
      end;
   gotoXY(1,15);
   writeln('Enter your choice');
   writeln('(e.g. dA1 for down from A1, aA1 for across from A1 ');
   write('and c for checking answer) : ');
   readln(choice);
   valid_input := true;
   if not (Upcase(choice[1]) in ['D', 'A', 'C']) then
     valid_input := false
   else
     if (Upcase(choice[1]) = 'C') then
      begin
         if (length(choice) <> 1) then
            valid_input := false
      end
   else
    if (length(choice) <> 3) then
       valid_input := false
    else
      if (Upcase(choice[2]) > 'E') or (Upcase(choice[2]) < 'A') then
         valid_input := false
     else
        begin
            val(choice[3],R,P);
           if P <> 0 then
              valid_input := false
        end;
   if valid_input = true then
     case Upcase(choice[1]) of
          'A' : begin
                C := ord(UpCase(choice[2]))-64;
                 write('Enter the word: ');
                 readln(wd);
                for j := 1 to length(wd) do
                   begin
                       if C+j-1 > 5 then
                          valid_input := false
                       else
                         if content[R,C+j-1] = '*' then
                            valid_input := false;
```

```
                    end;
                        if valid_input = true then
                            for j := 1 to length(wd) do
                               content[R,C+j-1] := UpCase(wd[j])
               end;
          'D' : begin
                 C := ord(Upcase(choice[2]))-64;
                  write('Enter the word: ');
                 readln(wd);
                  for i := 1 to length(wd) do
                      begin
                        if R+i-1 > 5 then
                          valid_input := false
                        else
                          if content[R+i-1,C] = '*' then
                            valid_input := false;
                      end;
                  if valid_input = true then
                     for i := 1 to length(wd) do
                        content[R+i-1,C] := UpCase(wd[i])
               end;
          'C' : begin
                  num_corr := 0;
                 for i := 1 to 5 do
                    for j := 1 to 5 do
                        if (content[i,j] = word[i,j]) and (content[i,j] <> '*') then
                          num_corr := num_corr + 1;
                         stop_game := true
              end;
      else
         valid_input := false
   end
end;

procedure Display_Result;
var
   i : integer;
begin
   for i := 15 to 20 do
     begin
       gotoXY(1,i);
       writeln('                                                                                ');
     end;
   gotoXY(1,15);
   writeln('You get ', num_corr, ' correct out of ', num_char, '.');
```

```
   writeln('Your correct percentage is ', num_corr/num_char*100:0:2, '%.') ;
   Player[index].Score := Player[index].Score + 2*num_corr;
   Player[index].Attempt := Player[index].Attempt + 2*num_char;
   store_users_info;
   readln
end;

begin                    {main body of crossword puzzle}
   quest_num := 1;
   randomize;
   repeat
      quest_no_str := chr(49 + random(4));
      Get_Words(quest_num);
      Get_Hints(quest_num);
      stop_game := false;
      Clrscr;
      Display_Table(5,5);
      Display_Hints;
      valid_input := true;
      repeat
         if valid_input = true then
            Display_Content(5,5)
         else
          begin
             write('Invalid input! Press <enter> to continue.');
             readln
          end;
         Get_Input
      until  stop_game;
      Display_Answer(5,5);
      Display_Result;
      quest_num := quest_num + 1;
      Player[index].Score := Player[index].Score + numCor;
      Player[index].Attempt := Player[index].Attempt + numQ;
      store_users_info;
      if quest_num <= max_no_q then
         begin
          writeln;
          write('Try again(Y/N)? ');
          readln(try_again)
         end
      else
         begin
          writeln;
          writeln('Press "N" to back to manu');
```

```
            try_again := 'N'
          end
      until Upcase(try_again) <> 'Y' ;
   end;

begin
   repeat
      textbackground(10);
      textcolor(4);
      Clrscr;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln('                Time: ',DateTimeToStr(now));
      writeln('                           English Game         ');
      writeln('                ======================================');
      writeln('                {                                 }');
      writeln('                {     1. English Guess Game         }');
      writeln('                {                                 }');
      writeln('                {     2. Crossword Puzzle           }');
      writeln('                {                                 }');
      writeln('                {     3. Quit                       }');
      writeln('                ======================================');
      repeat
        write('                  Enter your choice (1-3): ');
        readln(Selection);
      until Selection in [1..3];
      case Selection of
        1 : EngGuessGame;
        2 : CrossWPuzzle;
      end;
   until Selection = 3;
end;

{***********Liberal Studies Game***********}
Procedure LSGame;
var
   choice : integer;
   numQ, numCor : integer;
   continue : char;

   procedure LSMc;    {LSMc Game}
```

```
var
   NoOfQuest, L, NumOfCorrect : integer;
   MCQuestions, ChoOfQuest : array[1..20] of string;
   ModelAns : array[1..20] of char;
   MCQuestAns : char;
   MCQuests, MCChoice : string;
   MCFile : text;


   procedure Read_MCFile;
   var
      L : integer;
   begin
      assign(MCFile, 'LSMC.txt');
      reset(MCFile);
      readln(MCFile, NoOfQuest);
      for L := 1 to NoOfQuest do
      begin
         readln(MCFile, MCQuestions[L]);
         readln(MCFile, ChoOfQuest[L]);
         readln(MCFile, ModelAns[L])
      end;
    close(MCFile)
   end;


   procedure SelectMCQuest(var MCQuests, MCChoice : string);
   var
      m : integer;
   begin
      randomize;
      m := random(NoOfQuest)+1;
      MCQuests := MCQuestions[m];
      MCChoice := ModelAns[m];
   end;


begin   {Main body of LSMc Game}
   repeat
      Clrscr;
      Read_MCFile;
      writeln('There are ', NoOfQuest, ' questions in this test.');
      writeln;
      NumOfCorrect := 0;
      for L := 1 to NoOfQuest do
```

```
    begin
      SelectMCQuest(MCQuests, MCChoice);
      writeln('Question ', L);
      writeln(MCQuestions[L]);
      writeln(ChoOfQuest[L]);
      repeat
        write('Answer : ');
        readln(userans);
        if (userans > 'C') then
          userans := chr(ord(userans)-32);
        if (userans < 'A') and (userans > 'C') then
          writeln('Input A to C only. Try Again !');
      until (userans = 'A') or (userans = 'B') or (userans = 'C');
        if userans = ModelAns[L] then
          begin
            writeln('Right !');
            writeln;
            writeln;
            NumOfCorrect := NumOfCorrect + 1
          end
        else
          begin
            writeln('Wrong ! The answer is ', ModelAns[L]);
            writeln;
            writeln;
          end;
    end;
    writeln('You have ', NumOfCorrect, ' question(s) correct.');
    writeln('The percentage score is ', NumOfCorrect*100/NoOfQuest:5:2, '%');
    writeln('Rounded to 2 decimal places.');
    Player[index].Score := Player[index].Score + numCor;
    Player[index].Attempt := Player[index].Attempt + numQ;
    store_users_info;
    writeln;
    write('Try again(Y/y) to continue? ');
    readln(continue)
  until (continue <> 'Y') and (continue <> 'y');
end;    {End of LSMc Game}

procedure LSFill_in_the_blanks;   {LS Fill in the blanks}
var
  NoOfQuest, K, NumOfCorrect : integer;
  FBQuestions, ModelAns : array[1..50] of string;
  FBQuest, FBAns, userans : string;
  FBFile : text;
```

```
  procedure Read_FBFile;
  var
    K : integer;
  begin
    assign(FBFile, 'LSFB.txt');
    reset(FBFile);
    readln(FBFile, NoOfQuest);
    for K := 1 to NoOfQuest do
    begin
      readln(FBFile, FBQuestions[K]);
      readln(FBFile, ModelAns[K])
    end;
   close(FBFile)
  end;


  procedure SelectFBQuest(var FBQuest, FBAns : string);
  var
    m : integer;
  begin
    randomize;
    m := random(NoOfQuest)+1;
    FBQuest := FBQuestions[m];
    FBAns := ModelAns[m];
  end;

begin   {Main body of LSFB Game}
  repeat
    Clrscr;
    Read_FBFile;
    writeln('There are ', NoOfQuest, ' questions in this test.');
    writeln;

    NumOfCorrect := 0;
    for K := 1 to NoOfQuest do
    begin
      SelectFBQuest(FBQuest, FBAns);
      writeln('Question ', K);
      writeln(FBQuest);
      write('Ans : ');
      readln(userans);
      if FBAns = userans then
        begin
```

```
            writeln('Correct! ');
            writeln;
            writeln;
            NumOfCorrect := NumOfCorrect + 1 ;
          end
        else
          begin
            writeln('False! The answer is ', FBAns,' .');
            writeln;
            writeln;
          end;
      end;
      writeln('You have ', NumOfCorrect, ' question(s) correct.');
      writeln('The percentage score is ', NumOfCorrect*100/NoOfQuest:5:2, '%');
      writeln('Rounded to 2 decimal places.');
      Player[index].Score := Player[index].Score + numCor;
      Player[index].Attempt := Player[index].Attempt + numQ;
      store_users_info;
      writeln;
      write('Try again(Y/y) to continue? ');
      readln(continue)
    until (continue <> 'Y') and (continue <> 'y');
  end;


begin
  repeat
    textbackground(1);
    textcolor(7);
    Clrscr;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln;
    writeln('                Time: ',DateTimeToStr(now));
    writeln('                         LS Foundation Game        ');
    writeln('                ======================================');
    writeln('                {                                 }');
    writeln('                {     1. Multiple Choice Game       }');
    writeln('                {                                 }');
    writeln('                {     2. Fill in the blanks         }');
    writeln('                {                                 }');
    writeln('                {     3. Quit                      }');
```

```
    writeln('                ======================================');
    repeat
      write('                Enter your choice (1-3): ');
      readln(choice)
    until choice in [1..3];
    case choice of
      1 : LSMc;
      2 : LSFill_in_the_blanks;
    end;
  until choice = 3;
end;



{***********Maths Game***********}
procedure Mathsgame;
var
  choice : integer;
  Selection : integer;
  continue : char;

  procedure Addition_Game;
  var
    NumOfCorrect, NoOfQuest, i : integer;
    num1, num2, num3, noNum, Ans : integer;
    Userans : integer;
    time : integer;
  begin
    Clrscr;
    time:= gettickcount;
    NoOfQuest := 10;
    NumOfCorrect := 0;
    randomize;
    for i := 1 to NoOfQuest do
      begin
        num1 := random(100) + 1;
        num2 := random(100) + 1;
        num3 := random(100) + 1;
        noNum := random(2) + 2;
        if noNum = 2 then
          begin
            writeln('            Question : ', num1, ' + ', num2, ' = ');
            Ans := num1 + num2;
          end
        else
```

```
          begin
            writeln('            Question : ',num1, ' + ', num2, ' + ', num3, ' = ');
            Ans := num1 + num2 + num3;
          end;
        write('            Your answer is : ');
        readln(Userans);
        if Userans = Ans then
          begin
            writeln('Correct! Keep it up! ');
            NumOfCorrect := NumOfCorrect + 1;
            writeln;
            writeln;
          end
        else
          begin
            writeln('False! Never mind!');
            writeln('The correct answer is ', Ans);
            writeln('Try next question');
            writeln;
            writeln;
          end;
      end;
    time := (gettickcount-time) div 1000; { calculate elapsed time }
    writeln('You took ',time,' second(s) to complete.');
    writeln('You have ', NumOfCorrect, ' question(s) correct.');
    writeln('The percentage score is ', NumOfCorrect*100/NoOfQuest:5:2, '%');
    writeln('Rounded to 2 decimal places.');
    writeln;
    Player[index].Score := Player[index].Score + 2*NumOfCorrect;
    Player[index].Attempt := Player[index].Attempt + 2*NoOfQuest;
    store_users_info;
    repeat
      write('Try again(Y/y) to continue? ');
      readln(continue)
    until (continue <> 'Y') and (continue <> 'y');
end;

procedure Subtraction_Game;
var
    NumOfCorrect, NoOfQuest, i : integer;
    num1, num2, num3, noNum, Ans : integer;
    Userans : integer;
    time : integer;
begin
    Clrscr;
```

```
time:= gettickcount;
NoOfQuest := 10;
NumOfCorrect := 0;
randomize;
for i := 1 to NoOfQuest do
  begin
    num1 := random(100) + 1;
    num2 := random(50) + 1;
    num3 := random(100) + 1;
    noNum := random(2) + 2;
    if noNum = 2 then
      begin
        writeln('            Question : ', num1, ' - ', num2, ' = ');
        Ans := num1 - num2;
      end
    else
      begin
        writeln('            Question : ', num1, ' - ', num2, ' - ', num3, ' = ');
        Ans := num1 - num2 - num3;
      end;
    write('            Your answer is : ');
    readln(Userans);
    if Userans = Ans then
      begin
        writeln('Correct! Keep it up! ');
        NumOfCorrect := NumOfCorrect + 1;
        writeln;
        writeln;
      end
    else
      begin
        writeln('False! Never mind!');
        writeln('The correct answer is ', Ans);
        writeln('Try next question');
        writeln;
        writeln;
      end;
  end;
time := (gettickcount-time) div 1000; { calculate elapsed time }
writeln('You took ',time,' second(s) to complete.');
writeln('You have ', NumOfCorrect, ' question(s) correct.');
writeln('The percentage score is ', NumOfCorrect*100/NoOfQuest:5:2, '%');
writeln('Rounded to 2 decimal places.');
writeln;
Player[index].Score := Player[index].Score + 2*NumOfCorrect;
```

```
   Player[index].Attempt := Player[index].Attempt + 2*NoOfQuest;
   store_users_info;
   repeat
     write('Try again(Y/y) to continue? ');
     readln(continue)
   until (continue <> 'Y') and (continue <> 'y');
end;

procedure Multiplication_Game;
var
   NumOfCorrect, NoOfQuest, i : integer;
   num1, num2, num3, noNum, Ans : integer;
   Userans : integer;
   time : integer;
begin
   Clrscr;
   time:= gettickcount;
   NoOfQuest := 10;
   NumOfCorrect := 0;
   randomize;
   for i := 1 to NoOfQuest do
      begin
         num1 := random(20) + 1;
         num2 := random(20) + 1;
         num3 := random(20) + 1;
         noNum := random(2) + 2;
         if noNum = 2 then
            begin
               writeln('            Question : ', num1, ' * ', num2, ' = ');
               Ans := num1 * num2;
            end
         else
            begin
               writeln('            Question : ', num1, ' * ', num2, ' * ', num3, ' = ');
               Ans := num1 * num2 * num3;
            end;
         write('            Your answer is : ');
         readln(Userans);
         if Userans = Ans then
            begin
               writeln('Correct! Keep it up! ');
               NumOfCorrect := NumOfCorrect + 1;
               writeln;
               writeln;
               end
```

```
        else
           begin
             writeln('False! Never mind!');
             writeln('The correct answer is ', Ans);
             writeln('Try next question');
             writeln;
             writeln;
           end;
      end;
   time := (gettickcount-time) div 1000; { calculate elapsed time }
   writeln('You took ',time,' second(s) to complete.');
   writeln('You have ', NumOfCorrect, ' question(s) correct.');
   writeln('The percentage score is ', NumOfCorrect*100/NoOfQuest:5:2, '%');
   writeln('Rounded to 2 decimal places.');
   writeln;
   Player[index].Score := Player[index].Score + 2*NumOfCorrect;
   Player[index].Attempt := Player[index].Attempt + 2*NoOfQuest;
   store_users_info;
   repeat
      write('Try again(Y/y) to continue? ');
      readln(continue)
   until (continue <> 'Y') and (continue <> 'y');
end;

procedure Challenge_Mode;
var
   NumOfCorrect, NoOfQuest, i : integer;
   num1, num2, num3, opcode1, opcode2, Ans : integer;
   Userans : integer;
   time : integer;

function Numresult(num1, num2, opcode : integer) : integer;
begin
{0 represents + , 1 represents - , 2 represents *}
   case opcode of
      0 : Numresult := num1 + num2;
      1 : Numresult := num1 - num2;
      2 : Numresult := num1 * num2;
   end;
end;

function Mathsoperator(opcode : integer) : string[3];
begin
   case opcode of
   0 : Mathsoperator := ' + ';
```

```
    1 : Mathsoperator := ' - ';
    2 : Mathsoperator := ' * ';
    end;
  end;

  begin
    Clrscr;
    time:= gettickcount;
    NoOfQuest := 10;
    NumOfCorrect := 0;
    randomize;
    for i := 1 to NoOfQuest do
      begin
        num1 := random(10) + 1;
        num2 := random(10) + 1;
        num3 := random(10) + 1;
        opcode1 := random(3);
        opcode2 := random(3);
        if opcode2 = 2 then
          Ans := Numresult(num1, Numresult(num2, num3, opcode2), opcode1)
        else
          Ans := Numresult(Numresult(num1, num2, opcode1), num3, opcode2);
        writeln('            Question : ', num1, Mathsoperator(opcode1), num2,
Mathsoperator(opcode2), num3, ' = ');
        write('            Your answer is : ');
        readln(Userans);
        if Userans = Ans then
          begin
            writeln('Correct! Keep it up! ');
            NumOfCorrect := NumOfCorrect + 1;
            writeln;
            writeln;
          end
        else
          begin
            writeln('False! Never mind!');
            writeln('The correct answer is ', Ans);
            writeln('Try next question');
            writeln;
            writeln;
          end;
      end;
    time := (gettickcount-time) div 1000; { calculate elapsed time }
    writeln('You took ',time,' second(s) to complete.');
    writeln('You have ', NumOfCorrect, ' question(s) correct.');
```

```
    writeln('The percentage score is ', NumOfCorrect*100/NoOfQuest:5:2, '%');
    writeln('Rounded to 2 decimal places.');
    writeln;
    Player[index].Score := Player[index].Score + 2*NumOfCorrect;
    Player[index].Attempt := Player[index].Attempt + 2*NoOfQuest;
    store_users_info;
    repeat
      write('Try again(Y/y) to continue? ');
      readln(continue)
    until (continue <> 'Y') and (continue <> 'y');
  end;

begin
  repeat
    Clrscr;
    textbackground(3);
    textcolor(7);
    Clrscr;
    writeln;
    writeln;
    writeln;
    writeln('                  Time: ',DateTimeToStr(now));
    writeln('                             Maths_fun_game          ');
    writeln('                  ======================================');
    writeln('                  {                                 }');
    writeln('                  {     1. Addition Game             }');
    writeln('                  {                                 }');
    writeln('                  {     2. Subtraction Game          }');
    writeln('                  {                                 }');
    writeln('                  {     3. Multiplication Game       }');
    writeln('                  {                                 }');
    writeln('                  {     4. Challenge Mode            }');
    writeln('                  {                                 }');
    writeln('                  {     5. Quit                      }');
    writeln('                  {                                 }');
    writeln('                  {                                 }');
    writeln('                  ======================================');
    repeat
      write('                   Enter your choice (1-5): ');
      readln(choice)
    until choice in [1..5];
    case choice of
      1 : Addition_Game;
      2 : Subtraction_Game;
      3 : Multiplication_Game;
```

```
        4 : Challenge_Mode;
      end;
    until choice = 5;
end;


procedure display_user_score(index : integer);
var
  percent_score : real;
begin
  if Player[index].Attempt = 0 then
    percent_score := 0
  else
    percent_score := Player[index].score/Player[index].Attempt*100;
    clrscr;
    textbackground(0);
    textcolor(4);
    writeln;
    writeln('                          Your results               ');
    writeln('             Time: ',DateTimeToStr(now));
    writeln('             ============================================');
    writeln('                                      ');
    writeln('             ------------------------------------------- ');
    writeln;
    writeln('             Your User name       : ', Player[index].UserName);
    writeln;
    writeln('             Your name           : ', Player[index].EngName);
    writeln;
    writeln('             Form of Year         : ', Player[index].Form);
    writeln;
    writeln('             Total attempt score   : ', Player[index].Attempt);
    writeln;
    writeln('             Your total score     : ', Player[index].Score);
    writeln;
    writeln('             Your percentage score : ', percent_score:0:2);
    writeln('             ------------------------------------------- ');
    writeln;
    write('                     Press <Enter> to return. ');
    readln
end;


procedure UserOpinions;
var
  opinions : text;
```

```
  UserOp   : string;
  Username : string;
begin
  Clrscr;
  assign(opinions, 'opinions.txt');
  rewrite(opinions);
  writeln;
  writeln;
  writeln('                        Users" Opinions Region');
  writeln;
  writeln;
  writeln('         Your opinions is very important to us for improve the system!');
  writeln;
  writeln;
  write('             Please enter your user name : ');
  readln(Username);
  writeln;
  writeln;
  write('             What is your opinions? ');
  readln(UserOp);
  writeln(opinions, UserOp);
  close(opinions);
  writeln;
  writeln;
  writeln;
  writeln;
  writeln('                    Thanks you for your opinions!');
  writeln('                       Press Enter to continue.. ');
  readln;
end;

procedure Selectgame(index : integer);
var
  choice : integer;
begin
  repeat
    textbackground(0);
    Clrscr;
    textcolor(15);
    writeln;
    writeln;
    writeln('                       Welcome to play the learning gamer   ');
    writeln('                 Time: ',DateTimeToStr(now));
    writeln;
    writeln('                 ========================================');
```

```
    writeln('                  *                                *');
    writeln('                  *      1.  English Game            *');
    writeln('                  *                                *');
    writeln('                  *      2.  Mathemetics Game        *');
    writeln('                  *                                *');
    writeln('                  *      3.  LS Game                 *');
    writeln('                  *                                *');
    writeln('                  *      4.  Your Score              *');
    writeln('                  *                                *');
    writeln('                  *      5.  Your Opinions           *');
    writeln('                  *                                *');
    writeln('                  *      6.  Logout                  *');
    writeln('                  *                                *');
    writeln('                  ************************************');
    writeln;
    repeat
      write('                  Please enter your game choice: ');
      readln(choice);
    until choice in [1..6];
    case choice of
      1 : EnglishGame;
      2 : MathsGame;
      3 : LSGame;
      4 : display_user_score(index);
      5 : UserOpinions;
    end;
  until choice = 6
end;



{***********User Login System***********}
procedure UserRecords;
var
  UserAccount : text;
  UserIndex : integer;
  Choice : integer;
  UserNum : integer;
  UserPos : integer;
  attempt : array[1..max_user] of integer;
  score : array[1..max_user] of integer;


  function GetPWord : string;
  var
```

```
S : string;
C : Char;
begin
   S := '';
   repeat
      C := ReadKey;
      if (C <> #10) and (C <> #13) and (C <> #8) then
         begin
            S := S + C;
            write('*');
         end
      else if C = #8 then
         begin
            S[0] := Chr(Length(S) - 1);
            GotoXY(WhereX - 1, WhereY);
            write(' ');
            GotoXY(WhereX - 1, WhereY);
         end;
   until (C = #10) or (C = #13);
      GetPWord := S;
      writeLn;
end;


procedure CreateAccount;
var
   m, index : integer;
   found    : boolean;
   target   : string[30];
begin
   Clrscr;
   assign(UserAccount, 'userac.txt');
   append(UserAccount);
   numuser := numuser + 1;
   with Player[numuser] do
   begin
      writeln;
      writeln;
      writeln;
      writeln('                              Create Account');
      writeln;
      writeln;
      writeln;
      write('            Enter your user name : ');
      readln(UserName);
```

```
      writeln;
      writeln;
      found := false;
      m := 0;
      repeat
        m := m + 1;
        with Player[m] do
          if UserName = target then
            begin
              index := m;
              found := true
            end
      until found or (m = numuser);
      if found then
        begin
          writeln('Invalid user name! Please input again');
          with Player[index] do
            repeat
              begin
                write('            Enter your user name : ');
                readln(UserName);
                writeln;
                writeln;
              end;
            until found = false;
        end;
      write('            Enter your password : ');
      Password := GetPWord;
      writeln;
      writeln;
      write('            Enter your English Name: ');
      readln(EngName);
      writeln;
      writeln;
      write('            Enter of Year of form (e.g. "F3", "F6") : ');
      readln(Form);
      writeln(UserAccount, UserName);
      writeln(UserAccount, Password);
      writeln(UserAccount, EngName);
      writeln(UserAccount, Form);
      writeln(UserAccount, 0);
      writeln(UserAccount, 0);
      close(UserAccount);
    end;
end;
```

```
procedure login(var index : integer);
var
   userid, userpw    : string[30];
   found             : boolean;
   n, selection : integer;
begin
   repeat
      Clrscr;
      writeln;
      writeln;
      writeln;
      writeln;
      writeln('                                Login ');
      writeln;
      writeln;
      writeln;
      write('            Enter your user name: ');
      readln(userid);
      writeln;
      writeln;
      writeln;
      write('            Enter your password: ');
      userpw := GetPword;
      found := false;
      n := 0;
      repeat
         n := n + 1;
         with Player[n] do
            if (UserName = userid) and (Password = userpw) then
               begin
                  index := n;
                  found := true;
               end
      until found or (n > numuser);
      if found then
         begin
            Clrscr;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
            writeln;
```

```
          writeln;
          writeln;
          writeln;
          writeln('                        Hello! Welcome ', Player[index].EngName,'!');
          writeln;
          writeln;
          write('                       Press Enter to continue!');
          readln;
          Selectgame(index);
        end
     else
        begin
          Clrscr;
          writeln;
          writeln;
          writeln;
          writeln;
          writeln;
          writeln('                     Wrong UserName or Password!');
          writeln;
          writeln('                     1. login again?');
          writeln;
          writeln;
          writeln;
          writeln('                     2. Don"t have account? Create one now');
          writeln;
          writeln;
          writeln;
          write('Your selection? ');
          readln(selection);
         end;
   until found or (selection = 2);
        if selection = 2 then
          CreateAccount
end;

procedure UpdateAC(var count : integer);
var
   target   : string[30];
   found    : boolean;
   n, index : integer;
begin
   Clrscr;
   writeln;
   writeln;
```

```
writeln;
writeln;
writeln;
writeln;
writeln;
writeln;
writeln;
writeln('                         Update Your Account');
writeln;
writeln;
write('                  Enter your User Name: ');
readln(target);
found := false;
n := 0;
repeat
   n := n + 1;
   with Player[n] do
      if UserName = target then
        begin
          index := n;
          found := true
        end
until found or (n = numuser);
if found then
  begin
     Clrscr;
     writeln;
     writeln;
     writeln;
     writeln;
     writeln;
     writeln;
     writeln;
     writeln;
     writeln;
     writeln;
     write('                          Finding player record...');
     delay(2000);
     with Player[index] do
     begin
        Clrscr;
        writeln;
        writeln;
        writeln;
        writeln;
```

```
          writeln;
          writeln;
          writeln;
          writeln('                        Player record found!');
          writeln;
          writeln;
          write('             Enter your user name: ');
          readln(UserName);
          writeln;
          writeln;
          write('             Enter your password: ');
          GetPWord;
          writeln;
          writeln;
          store_users_info;
          writeln('                           Record updated!');
          writeln('                        Please Enter to return.');
          readln;
          writeln;
          writeln;
        end
      end
    else
      begin
        Clrscr;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln;
        writeln;
        write('                         Student name not found!');
        readln;
      end;
end;


begin    {Main of UserRecord}
  repeat
    textbackground(15);
    textcolor(0);
```

```
      clrscr;
      writeln;
      writeln;
      writeln;
      writeln('                    Welcome to play the learning gamer   ');
      writeln('                Time: ',DateTimeToStr(now));
      writeln;
      textcolor(3);
      writeln('                ======================================');
      writeln('                *                                    *');
      writeln('                *      1.  login your account        *');
      writeln('                *                                    *');
      writeln('                *      2.  Create your new account   *');
      writeln('                *                                    *');
      writeln('                *      3.  Update your account       *');
      writeln('                *                                    *');
      writeln('                *      4.  Quit                      *');
      writeln('                *                                    *');
      writeln('                ======================================');
      writeln;
      repeat
        write('                   Enter your choice (1-4): ');
        readln(choice)
      until choice in [1..4];
      case choice of
        1 : login(index);
        2 : CreateAccount;
        3 : UpdateAC(UserNum);
    end;
    until choice = 4;
end;


{***********Main Program***********}
begin
  ReadRecord;
  textcolor(10);
  writeln('
=========================================================================');
  writeln('   {  L      EEEEEEE    A    RRRRRR  NN    N IIIII NN    N   GGGGGG    }');
  writeln('   {  L      E         A A   R    R N N   N   I   N N   N  G         }');
  writeln('   {  L      EEEEEEE   AAAAA  RRRRRR  N  N  N   I   N  N  N G    GGGG  }');
  writeln('   {  L      E       A    A  RRRR    N   N N   I   N   N N  G     G   }');
  writeln('   {  LLLLLLL EEEEEEE A      A R   RRR N    NN IIIII N    NN   GGGGGG    }');
  textcolor(14);
```

```
  writeln('   {                                                          }');
  writeln('   {           GGGGGG       A     M      M EEEEEEE RRRRRR          }');
  writeln('   {          G           A A    MM    MM E      R     R          }');
  writeln('   {         G     GGGG   AAAAA   M M   M M EEEEEEE RRRRRR          }');
  writeln('   {          G     G    A     A  M  M M  M E      RRRR            }');
  writeln('   {           GGGGGG    A      A M   M   M EEEEEEE R   RRR         }');
  textcolor(11);
  writeln('   {    ============================================================ }');
  writeln('   {Time: ',DateTimeToStr(now), '                                    }');
  writeln('   {                                                          }');
  writeln('   {                                                          }');
  writeln('   {                                                          }');
  writeln('   {                                                          }');
  writeln('   {                  Please press ENTER to continue              }');
  writeln('
--------------------------------------------------------------------------');
  readln;
  UserRecords;
  Selectgame(index);
end.
```

7.2) Reference

1. Microsoft Corporation
2. Dev Pascal
3. Mr. Chu Kin Fung
4. Longman Hong Kong Education
5. http://freedailycrosswords.com/printable-crossword-puzzles/
6. http://en.wikipedia.org/wiki/Word_guessing
7. http://programmersheaven.com/categories/pasprog/?S=B20000
8. Google photo