

Content

Chapter1-Introduction.....	3
1.1- Background.....	3
1.2- Design.....	3
1.3- Data Flow.....	4
Chapter2-Implementation.....	5
2.1- Variable.....	5~6
2.2- Procedure.....	7~18
Mastermind.....	7~9
Maths Games.....	10~11
Word Guessing.....	12~14
Players' account related.....	15~19
Main program.....	19~21
2.3- User interface.....	22~28
Chapter 3-Testing and Evaluation.....	29~35
Chapte 4-Conclusion and future improvement.....	36~37
Appendix.....	38~75

Chapter 1-Introduction

1.1 Background

In the 21st century, technology is well developed. Other than traditional text book, ebook and online educating are the current trend. However, education is not the only way to seek knowledge. Gaming is another method. Taking myself as an example, I learnt lots of things from different games. I learn logistic thinking from mastermind, English vocab from video game, arithmetic from mathematics games. I think that using games to educate children is another way to arouse their potential. That's why I design this program

1.2 Design

In this program, I designed three games. One is mastermind and one is maths game. For the mastermind, player need to guess the combination of 4 numbers. The range of the numbers, the numbers of guessing can be chose by the player. If the player can guess the correct combination in limited chances, The player win. Otherwise, player lose and answer will be showed. For the maths game, player need to use three numbers which is random to form a formula with +, -, *. The player only get one chance. Neither the player is correct or not, model answer will be shown. For the word guessing game, an vocab will be siphon from the text file. The vocab will change into some * and player need to get hits but randomly press some words.

To make the program clear, I divided it in to serveal part.

1. Player login or register a new account.
2. After (register and) login, player will see the main menu and player can select different function such as playing different games, check score, change password or exit.
3. After playing the game, program will calculate the score of the player according to how many answer the player correct.
4. Player can choose continue or not. If not, the program will go back to main menu
5. Program will logout if player chose exit.

1.3 Data Flow

This program require the user.txt file and Words.txt file for running.

- i. **Users.txt**-This file is used to store the users' data including names,login id,password,gender,score.This file is able to input or change data through creating new account,changing password.

s0011234	Au Sham Ki, Bobby	M6
s0021234	Au Yue, Joanne	F0
s0031234	Chan Kai Bong	M0
s0041234	Chan Man Cheun	M0
s0051234	Chan Mei Ling	F0
s0061234	Chan Shui Wah, Shirley	F0
s0071234	Chan Tai Man	M0
s0081234	Chan Wai Yee, Wendy	F0
s0091234	Chan Yick Yee, Eliza	F0
s0101234	Chau Tung, Donnie	M0
s0111234	Cheung Chi Chung	M0
s0121234	Cheung Koo Ho	M0 (e.g.)

- ii. **Words.txt**-This file is used to store the vocab which are using in the Word Guessing game.It provide question(vocab)to the program.This file is not able to make adjustment.

apple
boy
dog
cat
orange
banana
dragon
stupid
zoo
zebra
ultraman
popular
horse
house
hand
head
home
feasibility
effectiveness
efficiency

Chapter 2-Implementation

In this chapter,I will introduce the implementation of the PuzzleGames system and function program.I will explain the meaning of variables,display the program code and display the user interface.

2.1-Variable (**Bold face** words refer to the name of the variable)

Choice-This variable is used to save the players choicein the main menu.

num_user-This variableis used to store the number of users which have already created an account.This will use for the login procedure.

user_pw-This variable is used for storing the marks of the player.

Digit-This variable is used to store the range of the Mastermind question's number which is input by the player.The minimum of the digit is 4 while the maximum is 9.

Chance-This variablr is used to store the chance of inputing the answer in Mastermind.This variable is input by the player.

i,j,k-This three variable is used for for-loop function.

Number1, Number2,Number3-This three variable is used to store the three number that random by the procedure in MathGame.

OperationLoc1,OperationLoc2-These variables are used to mark the position when copying the operations

Len-This variable is used to save the legth of the solution.It will use to measure the solution is valid or not.

CP-This variable is used to store the number of correct position in the guess of the player.

CN-This variable is used to store the number of correct number in the guess of the player.

Score-This variable is an array to save the score of the player earning.

Guess-This variable is used to store the number of guess that the player input before guessing the answer of mastermind

User_id-This variable is used to store the UserID that the player input.

User_pw- This variable is used to store the UserPW that the player input.

name-this variable is used to save the player names.

Ans-This is used to store the answer that the player input in Mastermind

Letter-This is used to store the letter that player answer in WordGuessing

Response-This is used to store the response of player in asking player continue or not in all the games

Operation1,Operation2-These two variables are used to store the operations which are copied from the answer that player input

Correct-This variable is used to store the result of the player and make response according to true or false.

Done-This variable is used to measure the player correct or use all the chances in Word Guessing.

Secret-This variable is used to store the word which is copied from the Words.txt as the question of Word guessing.

Solution-This variable is used to save the response which is input by player in answering the MathGame question

FirstNum, SecondNum,ThirdNum-These three variables are used to save the numbers generated in the procedure of MathGame.

2.2-Procedure & Function

Mastermind

1. DigitNum

This procedure is used to ask the player how many different digit he/she wants in the Mastermind

```
procedure DigitNum;  
begin  
  repeat  
    writeln('How many different digits(4~9) do you want?');  
    readln(num)  
  until (num >= 4) and (num <= 9);  
end;
```

2. guessing

This procedure is used to ask the player how many chance does he want in guessing the 'Mastermind' answer

```
procedure guessing;  
begin  
  repeat  
    write('How many guessing do you want to have (1 - 20)? ');  
    readln(chance)  
  until (chance >= 1) and (chance <= 20);  
end;
```

3. Gen

This procedure is used to generate the answer of the Mastermind

```
procedure Gen (var Code : string[4]);  
var A : integer; Strg : string;  
begin  
  Code := '';  
  for i := 1 to 4 do begin  
    repeat  
      A := random(num) + 1  
    until not Use[A];  
    Use[A] := true;  
    str(A, Strg);  
    Code := Code + Strg  
  end;  
end;
```

4. Check

This procedure is used to check how many CN,CP do the player's answer has

```
procedure Check (H : string[4]; var V, R : integer);
begin
  V := 0; R := 0;
  for j := 1 to 4 do
    for k := 1 to 4 do
      if (H[j] = Ans[k]) then
        if j = k then V := V + 1
        else R := R + 1
      end if;
    end for;
  end for;
end;
```

5. Display

This procedure is used to display how many CN,CP do the player's answer has

```
procedure Display;
begin
  for j := 1 to i do
    writeln('Guess ', j, '? ', Guess[j], CP[j]:10, 'P', CN[j]:2, 'N')
  end for;
end;
```


6. Mastermind

This is the main program of mastermind game. It runs the above procedure and displays the result. It also asks the player if they want to continue or not.

```
procedure Mastermind;
begin
  CLRSCR;
  randomize;
  repeat
    DigitNum;
    guessing;
    writeln;
    for i := 1 to num do Use[i] := false;
    for i := 1 to chance do begin
      CP[i] := 0; CN[i] := 0 end;
    Gen(Ans);
    i := 0; Correct := false;
    repeat
      i := i + 1;
      repeat
        write('Guess ', i, '? ');
        readln(Guess[i]);
        k := 0;
        for j := 1 to 4 do
          if (copy(Guess[i], j, 1) >= '1') and (copy(Guess[i], j, 1) <= chr
until (k = 4) and (length(Guess[i]) = 4);
        writeln;
        Check(Guess[i], CP[i], CN[i]);
        if CP[i] = 4 then Correct := true
        else Display;
      until (i = chance) or Correct;
      writeln;
      if Correct then
        writeln('You win!')
      else begin
        writeln('You lose!');
        writeln('Correct answer is ', Ans)
      end;
      writeln;
      write('Continue (y/n)? ');
      readln(Response);
      writeln;
    until (Response = 'n') or (Response = 'N');
  end;
```

Maths Games

7. Function F

This function is used for the following answer check and question check.

```
Function F(Number1,Number2:integer; operate:Char):integer ;
begin
  case operate of
    '+' : F := Number1+Number2;
    '-' : F := Number1-Number2;
    '*' : F := Number1*Number2;
  end
end;
```

8. NumberCheck

This procedure is used to ask player to give the solution,check and display it is correct or not.

```
procedure NumberCheck;
begin
  writeln('Please enter your solution');
  writeln('(an operation with '+','','-','*'.'/''is not allowed.)');
  write('? ');
  readln(Solution);
  Len := length(Solution);
  OperationLoc1 := 0;
  repeat
    OperationLoc1 := OperationLoc1 + 1;
    Operation1 := Solution[OperationLoc1];
  until (Operation1='+') or (Operation1='-') or (Operation1='*') or (OperationLoc1 > Len);
  OperationLoc2 := OperationLoc1;
  repeat
    OperationLoc2 := OperationLoc2 + 1;
    Operation2 := Solution[OperationLoc2];
  until (Operation2='+') or (Operation2='-') or (Operation2='*') or (OperationLoc2 > Len);

  if OperationLoc2 > Len then
    write('Missing or invalid answer!')
  else
    begin
      FirstNum := copy(Solution, 1, OperationLoc1-1);
      SecondNum := copy(Solution, OperationLoc1+1, OperationLoc2-OperationLoc1-1);
      ThirdNum:=copy(Solution,OperationLoc2+1,Len);
      val(FirstNum, Number1, Code1);
      val(SecondNum, Number2, Code2);
      val(ThirdNum, Number3, Code3);
      if (Code1 > 0) or (Code2 > 0) or (Code3 > 0) then
        write('Missing or invalid number(s)!')
      else
        begin
          if Operation2='*' then Answer := F(Number1,Number2*Number3,Operation1) else
            Answer := F(F(Number1,Number2,Operation1),Number3,Operation2);
          if Answer = 24
            then writeln('You are correct!')
            else writeln('You are wrong!');
        end;
      end;
    end;
  end;
```

9. GensAns

This procedure is used to generate the question with checking the feasibility of the question

```
procedure GenAns(var exp:string; var ans:integer);
var
x,y,z,op1,op2:integer;
Optor : array [1..3] of char;
begin
ans:=0;
optor[1]:='+';
optor[2]:='-';
optor[3]:='*';
Randomize;
repeat
x:=random(9)+1;
y:=random(9)+1;
z:=random(9)+1;
op1:=random(3)+1;
op2:=random(3)+1;
exp:= chr(x+48)+optor[op1]+chr(y+48)+optor[op2]+chr(z+48);
if op2 = 3 then ans:=F(x,y*z,optor[op1])
else Ans:= F(F(x,y,optor[op1]),z,optor[op2]);
until ans = 24;
writeln('Please form a operation with',x,',',y,',',z,'which the answer is 24.');
```

10.MathGame

This is the main program of MathGame which is used to call the above procedure,display the model answer and ask for continue or not.

```
procedure MathGame;
begin
CLRSCR;
repeat
GenAns(expression,answer);
NumberCheck;
writeln('The model ans is ',expression, '.');
writeln('Do you want to play again?(Y/y for continue,N/n for quit)');
readln(Response);
until (Response='n') or (Response='N');
end;
```

Word Guessing

11.SelectSecret

This part is used to read words.txt and select a vocab inside it as the question of the Word Guessing.

```
procedure SelectSecret(var Secret : string);
var word : array[1..50] of string;
infile : text;
NoOfWord : integer;
begin
  CLRSCR;
  assign(infile, 'Words.txt');
  reset(infile);
  i := 0;
  while not eof(infile) do
    begin
      i := i + 1;
      readln(infile, word[i])
    end;
  close(infile);
  NoOfWord := i;
  randomize;
  Secret := Word[random(NoOfWord)+1]
end;
```

12.FormQues

This part is used to change the secret which selected above to *.It also calculate the numbers of words of the secret and double it as the chance of guessing

```
procedure FormQues;
begin
  Solution := '';
  for i := 1 to length(Secret) do
    Solution := Solution + '*';
  Chance := 2 * length(Secret)
end;
```

.

13. LetterGuess

This part is used to ask for the answer which is a letter and compare it with the secret to find for any equal letter and change the corresponding * into that letter. At last, I will calculate the chance that the player left.

```
procedure LetterGuess;  
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);  
  Chance := Chance - 1;  
  writeln('Chane left = ', Chance)  
end;
```

14. This part is used to check the player answer all the correct letters or not.

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

15.WordGuessing

This is the main part of the Word Guessing Game.It call the 13.14. procedure and display the letter that have been answered correctly and the position of *.It also display the chance that players left.It will display 'You Win!!' and adding 1 point for the player if player answer all the letters of the secret.If not,it will display 'You lose!!'and show the correct answer.It will ask player continue or not after the above process done.

```
Procedure WordGuessing;
begin
  textcolor(13);
  repeat
    SelectSecret(Secret);
    FormQues;
    Done := false;
    writeln('The secret word to be guessed is ',Solution);
    writeln('Chance remaning = ',Chance) ;
    writeln;
    repeat
      LetterGuess;
      WinCheck
    until Done = true;
    if Solution = Secret then
      begin
        writeln('You win!!');
        score[user_index] := score[user_index] +1;
      end
    else
      begin
        writeln('You lose!!');
        writeln('The secret is ',secret);
      end;
    writeln;
    writeln('Continue?(Y/N) ');
    readln(response);
    until (Response='n') or (Response='N');
    readln
  end;
```

Players' account related

16.Function GetPW

This function is used to compare the password that player input and compare it with the users.txt's content.

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

17.Read_users_info

This part is used to read the users.txt's content including login ID,Password,Name,Gender and Score.

```
procedure read_users_info;  
var  
    i : integer;  
    f : text;  
begin  
    assign(f, 'users.txt');  
    reset(f);  
    i := 0;  
    while not eof(f) do  
        begin  
            i := i + 1;  
            readln(f, userid[i], userpw[i], name[i], gender[i], score[i]);  
        end;  
    num_user := i;  
    close(f)  
end;
```

18.Store_users_info

This part is used to create new account by the request of player.

```
procedure store_users_info;  
var i : integer;  
    Doc : text;  
begin  
    assign(Doc, 'users.txt');  
    rewrite(Doc);  
    for i := 1 to num_user do  
        begin  
            writeln(Doc, userid[i], userpw[i], name[i], gender[i], score[i]);  
        end;  
    close(Doc)  
end;
```

19.create_ac

This part is used to give instruction to input the required data while player is creating a new account.

```
procedure create_ac;  
var  
    loginid, username, password1, password2 : string;  
    gender_in : char;  
    id_OK, pw_OK : boolean;  
    i : integer;  
begin  
    clrscr;  
    writeln;  
    textcolor(Yellow);  
    writeln('          Your Name with more than 25 characters will be truncated.');    writeln('          Your Name cannot be changed after created.');    textcolor(LightGray);  
    write('          Please enter your Name (at most 25 char) : ');  
    readln(username);  
    repeat  
        writeln;  
        write('          Please enter your Gender (M or F) : ');  
        readln(gender_in)  
    until gender_in in ['M', 'F', 'm', 'f'];  
  
    id_OK := false;  
    repeat  
        clrscr;  
        writeln;  
        writeln('          Welcome ', username, '!');        writeln;  
        write('          Please enter your LoginID (4 char) : ');  
        readln(loginid);  
        if length(loginid) <> 4 then  
            begin  
                writeln;  
                writeln('          The length of login must be 4 char!');                write('          Press <Enter> to retry. ');  
                readln
```



```

        end
    else
        begin
            id_OK := true;
            for i := 1 to num_user do
                if loginid = userid[i] then
                    begin
                        writeln;
                        writeln('          This LoginID has been used!');
                        write('          Press <Enter> to enter again. ');
                        readln;
                        id_OK := false
                    end
                end
            end
        until id_OK;

        pw_OK := false;
        repeat
            clrscr;
            writeln;
            writeln('          Welcome ', username, '!');
            writeln;
            writeln('          Your LoginID is ', loginid);
            writeln;
            write('          Please enter your password (4 char) : ');
            password1 := GetPword;
            if length(password1) <> 4 then
                begin
                    writeln;
                    writeln('          The length of password must be 4 char!');
                    write('          Press <Enter> to retry. ');
                    readln
                end
            else
                begin
                    writeln;
                    write('          Please enter your password again      : ');
                    password2 := GetPword;
                    if password1 <> password2 then
                        begin
                            writeln;
                            writeln('          The passwords do not match!');
                            write('          Press <Enter> to retry. ');
                            readln
                        end
                    else
                        begin
                            num_user := num_user + 1;
                            username := username + '          ';
                            username := copy(username, 1, 25);

                            userid[num_user] := loginid;
                            userpw[num_user] := password1;
                            name[num_user] := username;
                            gender[num_user] := gender_in;
                            score[num_user] := 0;

                            store_users_info;

                            pw_OK := true;
                            writeln;
                            writeln('          Your account is created successfully. ');
                            write('          Press <Enter> to return. ');
                            readln
                        end
                    end
                until pw_OK
            end;
        end;

```

20.display_user_score

This part is used to display the login id,name and score of the player.

```
procedure display_user_score(user_index : integer);
begin
    textcolor(green);
    clrscr;
    writeln;
    writeln('          Your results');
    writeln;
    writeln('          ***** ');
    writeln;
    writeln('          Your UserID          : ', userid[user_index]);
    writeln;
    writeln('          Your name            : ', name[user_index]);
    writeln;
    writeln('          Your total score      : ', score[user_index]);
    writeln('          ***** ');
    writeln;
    write('          Press <Enter> to return. ');
    readln
end;
```

21.change_password

This part is used to change the password if player requested.

```
procedure change_password(user_index : integer);
var
    oldpass, newpass1, newpass2 : string;
    pwchanged : boolean;
begin
    pwchanged := false;
    repeat
        clrscr;
        writeln;
        write('          Please enter your old password          : ');
        oldpass := GetPword;
        if oldpass <> userpw[user_index] then
            begin
                writeln;
                writeln('          Incorrect old password!');
                write('          Press <Enter> to try again. ');
                readln
            end
        else
            begin
                writeln;
                write('          Please enter your new password (4 char) : ');
                newpass1 := GetPword;
                if length(newpass1) <> 4 then
                    begin
                        writeln;
                        writeln('          The length of password must be 4!');
                        write('          Press <Enter> to retry. ');
                        readln
                    end
                else
                    begin
                        writeln;
                        write('          Please enter your new password one more time : ');
```

```

        newpass2 := GetPword;
        if newpass1 <> newpass2 then
            begin
                writeln;
                writeln('                The new passwords do not match!');
                write('                Press <Enter> to retry. ');
                readln
            end
        else
            begin
                userpw[user_index] := newpass1;
                store_users_info;
                pwchanged := true;
                writeln;
                writeln('                Password has been changed. ');
                write('                Press <Enter> to return. ');
                readln
            end
        end
    end
end
until pwchanged
end;

```

Main program

22.main menu

This part is used to display the function of this program including Mastermind, MathGame, WordGuessing, CheckScore, ChangePassword and Exit. Related procedure or process will be run according to player choice. This procedure will be run automatically after player login.

```

Procedure main_menu(user_index : integer);
begin
repeat
    textcolor(Yellow);
    clrscr;
    writeln;
    writeln('                Welcome to Puzzle Games ');
    writeln('                Please choose:');
    writeln;
    writeln('                ~~~~~~');
    writeln('                1.MasterMind');
    writeln;
    writeln('                2.MathGame');
    writeln;
    writeln('                3.WordGuessing');
    writeln;
    writeln('                4.CheckScore');
    writeln;
    writeln('                5.ChangePassword');
    writeln;
    writeln('                6.Exit');
    writeln;
    writeln('                ~~~~~~');
    write('                What is your choice: ');
    readln(Choice);
    writeln;
case choice of
        1 : Mastermind;
        2 : MathGame;
        3 : WordGuessing;
        4 : display_user_score(user_index);
        5 : change_password(user_index);

```

```

        end;
until choice = 6;
end;

```

23.login

This part of the program is require player to login their own account.It will run the above procedure which are related to logging in.

```

procedure login(var user_index : integer);
var
    loginid, password : string;
    found : boolean;
    i : integer;
begin
    clrscr;
    writeln;
    writeln;
    writeln('                Welcome to Puzzle Game                ');
    writeln;
    writeln('                *****                ');
    writeln('                *                *                ');
    writeln('                *                *                ');
    writeln('                *                *                ');
    writeln('                *                *                ');
    writeln('                *****                ');
    writeln;
    write('                LoginID: ');
    readln(loginid);
    writeln;
    write('                Password : ');
    password := GetPword;
    writeln;
    writeln;
    found := false;
    i := 0;
    while (i < num_user) and (not found) do
    begin
        i := i + 1;
        if (loginid = userid[i]) and (password = userpw[i]) then
        begin
            found := true;
            user_index := i;
        end
    end
    end;
    if not found then
    begin
        user_index := 0;
        textcolor(LightBlue);
        writeln('':20,'> > > Invalid UserID or Password!');
        write('':20,'> > > Press <Enter> to return. ');
        textcolor(green);
        readln
    end
    else
        main_menu(user_index)
    end;
end;

```

24.first_page

This part is the first page of the program.It used to let the player choosing from login or creat account.

```
procedure first_page;
begin
    clrscr;
    writeln;
    writeln('                Welcome to Puzzle game ');
    writeln('                Please choose:');
    writeln;
    writeln('=====');
    writeln('                1. Login');
    writeln;
    writeln('                2. Create account');
    writeln;
    writeln('=====');
    write('                Enter your choice: ');
    readln(choice);
    writeln;
    case choice of
        1 : login(user_index);
        2 : create_ac;
    end;
end;
```

25.Main program

This is the main part of the program and it used to call the first page procedure.

```
begin
read_users_info;
    repeat
        first_page;
    until false;
    readln
end.
```

2.3-User interface

I use Dev-pascal for program coding

The name of my program is PuzzleGame

Here is the example:

Firstly, this is the first page of the program. This is the welcome page. It lets the player choose from logging in or creating an account.

```

Welcome to Puzzle game
Please choose:

=====

1. Login

2. Create account

=====

Enter your choice:
```

Creating account

After entering '2' in the first page, the program will go to the creating account page. All of the creating account procedure will be run. It will first ask for the player's name, then gender.

```

Your Name with more than 25 characters will be truncated.
Your Name cannot be changed after created.
Please enter your Name (at most 25 char) : TimothyLeong

Please enter your Gender (M or F) : M
```

After that,the program will ask for the player's LoginID and password for the login process.

```
Welcome TimothyLeong!

Your LoginID is 1234

Please enter your password (4 char) : ****

Please enter your password again    : ****

Your account is created successfully.
Press <Enter> to return.
```

After entering all the information,pressing enter will back to the first page.

Logging in

In this process,player need to enter his LoginID and password

```
                Welcome to Puzzle Game

*****
*                                           *
*                               LOGIN          *
*                                           *
*****

LoginID: 1234

Password : ****
```

If LoginID or password is wrong,this message will be shown

```
                Welcome to Puzzle Game

*****
*                                           *
*                               LOGIN          *
*                                           *
*****

LoginID: 1234

Password : ****

> > > Invalid UserID or Password!
> > > Press <Enter> to return.
```

Pressing enter will go back to the first page and player need to do all the process again

Main menu

In case the player has login successful, the main menu will be shown. It look like this:

```
      Welcome to Puzzle Games
      Please choose:

      ~~~~~

      1.MasterMind
      2.MathGame
      3.WordGuessing
      4.CheckScore
      5.ChangePassword
      6.Exit

      ~~~~~

      What is your choice:
```

Player can use different function of this program by input different number.

1. Mastermind-Enter '1' will go into the Mastermind game.
2. MathGame-Enter '2' will go into the MathGame.
3. WordGuessing-Enter '3' will go into the WordGuessing game.
4. Checkscore-Enter '4' will go into the score checking pages which will show players' name, login ID and score.
5. ChangePassword-Enter '5' will go into the password changing process.
6. Exit-Enter '6' will logout and go back to the first page of the program

Mastermind

First, the program will ask the basic setting of the game. Player need to answer them according to the difficulties that he wants.

```
How many different digits(4~9) do you want?  
4  
How many guessing do you want to have (1 - 20)? 20
```

After this, the program will automatically generate a string with 4 different numbers according to the setting.

```
Guess 1? 1234      1P 3N  
Guess 2? 1324      0P 4N  
Guess 3? 4214      0P 4N  
Guess 4? 4321      1P 3N  
Guess 5? 4213      0P 4N  
Guess 6? 3214      0P 4N  
Guess 7? 3124      0P 4N  
Guess 8? 1432      2P 2N  
Guess 9? 1423      1P 3N  
Guess 10? 1342     0P 4N  
Guess 11? 2431  
  
You win!  
Continue (y/n)?
```

After each guess, the program will show how many number did the player input correctly. In this game, there have two different 'correct'. One is correct position (P) one is correct number (N). When player have 4P, Player will win and get one score. If player can guess the correct answer in the limited chance, player will lose and correct answer will be shown.

```
You lose!  
Correct answer is 2134  
Continue (y/n)?
```

The game will keep continue if player input 'y' in the continue. Input 'n' will back to the main menu.

MathGame

First, the program will automatically generate 3 numbers randomly. Player needs to use these three numbers to form a formula which the answer is 24. To prevent those three numbers to form a formula which the answer is 24, the program will form all possible formulas at first. It will display as the question if it can form a formula which the answer is 24, otherwise, it will generate three numbers again.

```
Please form a formula with 2,6,2 which the answer is 24.  
Please enter your solution  
(an operation with '+', '-', '*', '/' is not allowed.)  
? 2*6*2  
You are correct!  
The model answer is 2*6*2.  
Do you want to play again?(Y/y for continue, N/n for quit)
```

The reason that why it shows the model answer even the player's answer is correct is because it may have lots of combinations to form the formula. It is only for the reference.

WordGuessing

First, the program will automatically copy a vocab from the Words.txt file to form the question. The vocab will turn to some * according to the letters of the vocab. The chance of guessing is double of the number of letters.

```
The secret word to be guessed is *****  
Chance remaining = 10  
  
Please input the letter you guess:
```

Player needs to randomly input a letter to find hints.

```

Chane left = 7
Please input the letter you guess: g
*****
Chane left = 6
Please input the letter you guess: e
*****e
Chane left = 5
Please input the letter you guess: g
*****e
Chane left = 4
Please input the letter you guess: u
**u*e
Chane left = 3
Please input the letter you guess:

```

After each guessing, the chance will -1 and the letter that the player guessed will replace the * if it is the correct one. This message and answer will be displayed if player fail to guess the vocab in the limited chance.

```

Chane left = 0
You lose!!
The secret is house
Continue?(Y/N)

```

This is the situation if the player success.

```

You win!!
Continue?(Y/N)

```

CheckScore

In this page, the program will display the users' UserID, Name and total score.

```

Your results
*****
Your UserID      : 1234
Your name       : TimothyLeong
Your total score : 2
*****
Press <Enter> to return.

```

ChangePassword

In this process, player first need to enter the old password. If the password is incorrect, this message will be shown.

```
Please enter your old password      : ****  
  
Incorrect old password!  
Press <Enter> to try again.
```

If the password is correct, player can enter a new password. Player need to enter the new password one more time for the confirmation. If these two passwords are not the same, this message will be shown and the new password will not be used.

```
The new passwords do not match!  
Press <Enter> to retry.
```

If these two new passwords matched, this message will be shown and password will be changed.

```
Password has been changed.  
Press <Enter> to return.
```

Exit

Input '6' into the main menu will go back to the first page. To this, the program will be repeated and this program is ended.

Chapter 3-Testing and Evaluation

This program is tested by inputting some incorrect or invalid decision, choices or data to check for errors.

First page

In the first page, I input 3 in to the choice.

```
      Welcome to Puzzle game
      Please choose:

=====

      1. Login
      2. Create account

=====

      Enter your choice: 3
```

Nothing would happen until I input 1 or 2.

Creating account

In the creating account page, I input abcdefghijklmnopqrstuvwxyz 26 letters as the name which the limitation is 25. This message will be shown.

```
      Your Name cannot include more than 25 characters.
      Your Name cannot be changed after created.
      Please enter your Name (at most 25 char) : abcdefghijklmnopqrstuvwxyz
      You name is too long! Please make adjustment.
      Press <Enter> for retrying
```

Then I input G as the gender,

It will ask the question again until the gender is M or F.

```
      Your Name cannot include more than 25 characters.
      Your Name cannot be changed after created.
      Please enter your Name (at most 25 char) : Timothy

      Please enter your Gender (M or F) : G

      Please enter your Gender (M or F) :
```

After that,I try to input 5 char as the LoginID.This message will be shown.

```
Welcome Timothy!  
  
Please enter your LoginID <4 char> : 12345  
  
The length of login must be 4 char!  
Press <Enter> to retry.
```

I aslo try the repeated LoginID which has already used by other players.This message will be shown.

```
Welcome Timothy!  
  
Please enter your LoginID <4 char> : 1999  
  
This LoginID has been used!  
Press <Enter> to enter again.
```

Then I try to input 5 char into the password.This message has been shown.

```
Welcome Timothy!  
  
Your LoginID is 7890  
  
Please enter your password <4 char> : *****  
  
The length of password must be 4 char!  
Press <Enter> to retry.
```

I used 1234 as the password.

I try to enter a different password again.This message has been shown.

```
Welcome Timothy!  
  
Your LoginID is 7890  
  
Please enter your password <4 char> : ****  
  
Please enter your password again : ****  
  
The passwords do not match!  
Press <Enter> to retry.
```

All the process of creating account has been checked.

Login

First ,I input a LoginID which hasn't input to the users.txt yet and a wrong password.This message has been shown.

```

Welcome to Puzzle Game

*****
*                                     *
*                               LOGIN *
*                                     *
*****

LoginID: 123456

Password : *****

> > > Invalid UserID or Password!
> > > Press <Enter> to return.
```

Main Menu

In this page,I try to input a number which is not 1 to 6.Main menu is still displayed until the choice is between 1 to 7.

```

Welcome to Puzzle Games
Please choose:

~~~~~

1.MasterMind
2.MathGame
3.WordGuessing
4.CheckScore
5.ChangePassword
6.Exit

~~~~~

What is your choice: 7
```

Mastermind

First, I input a number which is not between 4 to 9. The program will keep asking the same question again until the number is between 4 to 9.

```
How many different digits<4~9> do you want?  
3  
How many different digits<4~9> do you want?
```

Second, I input a number which is not between 1 to 20 for the guessing. The program will keep asking the same question again until the number is between 1 to 20.

```
How many different digits<4~9> do you want?  
3  
How many different digits<4~9> do you want?  
4  
How many guessing do you want to have <1 - 20>? 0  
How many guessing do you want to have <1 - 20>?
```

Third, I input 5 numbers into the guess. The guessing will not reduce and it will keep asking until there are only 4 numbers.

```
How many different digits<4~9> do you want?  
3  
How many different digits<4~9> do you want?  
4  
How many guessing do you want to have <1 - 20>? 0  
How many guessing do you want to have <1 - 20>? 10  
  
Guess 1? 12345  
Guess 1?
```

Forth, I try to input the numbers which are out of range of the digit that the player input before. The guessing will not reduce and it will keep asking until they are inside the limited digit.

MathGame

First I tried to input a fomula which include '/'.In this program,'/' is not allowed so the program will show this massage.

```
Please form a operation with 3,6,6 which the answer is 24.
Please enter your solution
<an operation with '+', '-', '*', '/' is not allowed.>
? 3+3/6
Missing or invalid answer!The model ans is 3*6+6.
Do you want to play again?<Y/y for continue,N/n for quit>
```

Then I tried to input a formula which include three numbers that the question isn't given.I discover that the program still seek this as correct,so I need to make improvement of my program.

I change the procedure **8.NumberCheck** into this:

```
begin
  FirstNum := copy(Solution, 1, OperationLoc1-1);
  val(FirstNum,i);
  if (i <> x) and (i <> y) and (i <> z)
  then writeln('Invalid answer!')
  else begin
    SecondNum := copy(Solution, OperationLoc1+1, OperationLoc2-OperationLoc1-1);
    val(SecondNum,i);
    if (i <> x) and (i <> y) and (i <> z)
    then writeln('Invalid answer!')
    else begin
      ThirdNum:=copy(Solution,OperationLoc2+1,Len);
      val(ThirdNum,i);
      if (i <> x) and (i <> y) and (i <> z)
      then writeln('Invalid answer!');
    end;
  end;
end;
```

While copying the integers from the formula that the player input,the program will compare the integers with x,y,z which are the integer generate by the program as the question.If the integer that the player input is not one of the integer that the program generate,the program will define it as a invalid answer.Here is the example:

```
Please form a formula with 2,2,6 which the answer is 24.
Please enter your solution
<an operation with '+', '-', '*', '/' is not allowed.>
? 24+0+0
Invalid answer!
Missing or invalid number(s)!The model ans is 2*2*6.
Do you want to play again?<Y/y for continue,N/n for quit>
```

This can solve the problem which is player input 24+0+0 to fake the program and get score easily.

WordGuessing

In this game, I use different method to find bugs. First, I input the same letters twice. The program seek that two letters as one and check it is correct or not.

```
The secret word to be guessed is *****
Chance remaning = 26

Please input the letter you guess: e
e**e*****e**
Chane left = 25
Please input the letter you guess: ff
effe*****e**
Chane left = 24
Please input the letter you guess: aaaaaaaaaaaaaa
effe*****e**
Chane left = 23
Please input the letter you guess:
```

Then I input different letters in the same guessing, it only take the first letter as the answer.

```
The secret word to be guessed is *****
Chance remaning = 26

Please input the letter you guess: e
e**e*****e**
Chane left = 25
Please input the letter you guess: ff
effe*****e**
Chane left = 24
Please input the letter you guess: aaaaaaaaaaaaaa
effe*****e**
Chane left = 23
Please input the letter you guess: effectiveness
effe*****e**
Chane left = 22
Please input the letter you guess: ctive
effec*****e**
Chane left = 21
Please input the letter you guess:
```

Only taking the first letter as the guessing can prevent player randomly input lots of letters and find the answer.

Change Password

First, I tried to enter a wrong old password, this message has been displayed.

```
Please enter your old password      : ****  
  
Incorrect old password!  
Press <Enter> to try again.
```

Player need to press <Enter> and input the password again. It won't go to the next step before player input a correct password.

Second, after input a correct password, the program will ask the player to input a new password. If turning off the program in this moment, no adjustment will be made.

In this part I try to input a password which has 5 digit. This message has been shown

```
Please enter your old password      : ****  
  
Please enter your new password <4 char> : *****  
  
The length of password must be 4!  
Press <Enter> to retry.
```

For security concern, player need to enter the old password one more time if he enter a invalid new password.

If player input a valid password in this moment, player need to input it one more time to prevent any misclicking and make the player input a password that he don't want. If player enter a different password compare with the first time, this message will be shown

```
Please enter your old password      : ****  
  
Please enter your new password <4 char> : ****  
  
Please enter your new password one more time      : ****  
  
The new passwords do not match!  
Press <Enter> to retry.
```

Player need to do all the above step again if he input two different password

Chapter 4-Conclusion and future improvement

4.1-Conclusion

I believe using this program can meet my target-learning from gaming. Just like what I mention in the introduction, Gaming is a good way to learn different knowledge or training different skills. Take the hottest games in these few years as an example, Battlefield which is a first-person shooting game can train players' reaction, League of Legends which is a Multiplayer Online Battle Arena game can train players' teamworking skill. This program included three games, each of them have different use.

For the Mastermind, this game is concerning players' inference. Players need to first guessing for a random combination. Then, player need to prove the answer with those hints through think logically. This game can train players' logical think. A good logical think can help people prevent lots of troubles. For example, students can eaier prove the meaning of the passage in the reading exam, easier to solve the maths problem. Workers can easier to think about the logic of someone opinion. This shows that logical thinking is very useful and it benefit our daily life. It is worth to spend some times in playing mastermind to train the logical thinking.

For the MathGame, it mostly benefit the students. This game require player to form a formula by themselves. The original concept of this game is Mathematical Olympiad. It's target is training player thinking, bring up player's hobby in maths. This is the main point of the MathGame. I wish I can use this program to make player loving maths instead of hating it.

For the word guessing game, the main target is let the player learn more vocabs. Through guessing the vocab, player will attract by the funny of this game. They will hope to learn more to improve their knowledge. I hope this program can become a channel to stimulate players' study interest.

Although the program isnt perfect, I still want to use this program as a platform to benefit the players. I want to use this program to prove that gaming isnt a bad thing.

4.2-Future improvement

While creating the program,I discover that it is not a easy thing to create from 0 to 1.And at last I found out that my program need lots of improvement.

The biggest problem is there are some loopholes in each game.For example,in the maths game,before improvement,players can use cheats and fake the system.There may have other loopholes or bugs in the program.I willl keep improveing the program and willing to minimize the loopholes and bugs.

Another problem is that the numbers of game is not enough.This program only have 3 games,I don't think it is enough to bring up players' study interest.In the future,I will create more games and put them into the program.I want to make the content of this program become more abundant.

Appendix

program PuzzleGames;

Uses CRT;

const

max_user = 1000;

var

choice:integer;

num_user, user_index ,digit, chance, i, j, k ,Number1,

Number2,Number3, Answer :integer;

OperationLoc1,OperationLoc2, Len, Code1, Code2 , Code3 :

integer;

CP, CN : array [1..20] of integer;

score : array[1..max_user] of integer;

Guess : array [1..20] of string;

userid : array[1..max_user] of string[4];

userpw : array[1..max_user] of string[4];

name : array[1..max_user] of string[25];

Ans : string[4];

Use : array [1..52] of boolean;

gender : array[1..max_user] of char;

Letter,Response,Operation1,Operation2:Char;

Correct,Done : boolean;

**Secret,Solution, FirstNum, SecondNum,ThirdNum,expression:
string;**

{Mastermind}

procedure DigitNum;

begin

repeat

writeln('How many different digits(4~9) do you want?');

readln(digit)

until (digit >= 4) and (digit <= 9);

end;

procedure guessing;

begin

```
repeat
    write('How many guessing do you want to have (1 - 20)?
');
    readln(chance)
until (chance >= 1) and (chance <= 20);
end;
```

```
procedure Gen (var Code : string[4]);
var A : integer;  Strg : string;
begin
    Code := "";
    for i := 1 to 4 do begin
        repeat
            A := random(digit) + 1
        until not Use[A];
        Use[A] := true;
        str(A, Strg);
        Code := Code +  Strg
    end;
```


end;

procedure Check (H : string[4]; var V, R : integer);

begin

V := 0; R := 0;

for j := 1 to 4 do

for k := 1 to 4 do

if (H[j] = Ans[k]) then

if j = k then V := V + 1

else R := R + 1

end;

procedure Display;

begin

for j := 1 to i do

writeln('Guess ', j, '? ', Guess[j], CP[j]:10, 'P', CN[j]:2,

'N')

end;

```

procedure Mastermind;

begin

    randomize;

    repeat

        CLRSCR;

        textcolor(10);

        DigitNum;

        guessing;

        writeln;

        for i := 1 to digit do Use[i] := false;

        for i := 1 to chance do begin

            CP[i] := 0; CN[i] := 0 end;

        Gen(Ans);

        i := 0; Correct := false;

        repeat

            i := i + 1;

            repeat

                write('Guess ', i, '? ');

```

```

        readln(Guess[i]);

        k := 0;

        for j := 1 to 4 do

            if (copy(Guess[i], j, 1) >= '1') and
(cop
y(Guess[i], j, 1) <= chr(digit + 48)) then k:= k + 1

            until (k = 4) and (length(Guess[i]) = 4);

        writeln;

        Check(Guess[i], CP[i], CN[i]);

        if CP[i] = 4 then Correct := true

            else Display;

        until (i = chance) or Correct;

        writeln;

        if Correct then

            begin

                writeln('You win!');

                score[user_index] := score[user_index] +1;

            end

        else begin

            writeln('You lose!');

```

```

        writeln('Correct answer is ', Ans)

    end;

    writeln;

    write('Continue (y/n)? ');

    readln(Response);

    writeln;

    until (Response = 'n') or (Response = 'N');

end;

{Maths Games}

Function F(Number1,Number2:integer; operate:Char):integer ;

begin

    case operate of

        '+' : F := Number1+Number2;

        '-' : F := Number1-Number2;

        '*' : F := Number1*Number2;

    end

end;

end;

```

```

procedure NumberCheck;

begin

    textcolor(Yellow);

    writeln('Please enter your solution');

    writeln('(an operation with "+", "-", "*", "/" is not allowed.)');

    write('? ');

    readln(Solution);

    Len := length(Solution);

    OperationLoc1 := 0;

    repeat

        OperationLoc1 := OperationLoc1 + 1;

        Operation1 := Solution[OperationLoc1];

        Until (Operation1='+') or (Operation1='-') or (Operation1='*')
or (OperationLoc1 > Len);

        OperationLoc2 := OperationLoc1;

        repeat

            OperationLoc2 := OperationLoc2 + 1;

            Operation2 := Solution[OperationLoc2];

```

Until (Operation2='+') or (Operation2='-') or (Operation2='*')
or (OperationLoc2 > Len);

if OperationLoc2 > Len then

write('Missing or invalid answer!')

else

begin

FirstNum := copy(Solution, 1, OperationLoc1-1);

SecondNum := copy(Solution, OperationLoc1+1,

OperationLoc2-OperationLoc1-1);

ThirdNum:=copy(Solution,OperationLoc2+1,Len);

val(FirstNum, Number1, Code1);

val(SecondNum, Number2, Code2);

val(ThirdNum, Number3, Code3);

if (Code1 > 0) or (Code2 > 0) or (Code3 > 0) then

write('Missing or invalid number(s)!')

else

begin

```

        if Operation2='*' then Answer :=
F(Number1,Number2*Number3,Operation1)else

        Answer :=

F(F(Number1,Number2,Operation1),Number3,Operation2);

        if Answer = 24

        then

        begin

        writeln('You are correct!');

        score[user_index] := score[user_index] +1;

        end

        else writeln('You are wrong!');

        end;

    end;

end;

procedure GenAns(var exp:string; var ans:integer);

var

x,y,z,op1,op2:integer;

Optor : array [1..3] of char;

```

```

begin

  ans:=0;

  optor[1]:='+';

  optor[2]:='-';

  optor[3]:='*';

  Randomize;

  repeat

    x:=random(9)+1;

    y:=random(9)+1;

    z:=random(9)+1;

    op1:=random(3)+1;

    op2:=random(3)+1;

    exp:=

chr(x+48)+optor[op1]+chr(y+48)+optor[op2]+chr(z+48);

    if op2 = 3 then ans:=F(x,y*z,optor[op1])

    else Ans:= F(F(x,y,optor[op1]),z,optor[op2]);

  until ans = 24;

  textcolor(10);

```



```
writeln('Please form a formula with ',x,',',y,',',z,' which the  
answer is 24.');
```

```
end;
```

```
procedure MathGame;
```

```
begin
```

```
  textcolor(9);
```

```
  repeat
```

```
    CLRSCR;
```

```
    GenAns(expression,answer);
```

```
    NumberCheck;
```

```
    writeln('The model ans is ',expression,'.');
```

```
    writeln('Do you want to play again?(Y/y for continue,N/n for  
quit)');
```

```
    readln(Response);
```

```
    until(Response='n') or (Response='N');
```

```
end;
```

```
procedure SelectSecret(var Secret : string);  
  
var word : array[1..50] of string;  
  
infile : text;  
  
NoOfWord : integer;  
  
begin  
  
    CLRSCR;  
  
    assign(infile, 'Words.txt');  
  
    reset(infile);  
  
    i := 0;  
  
    while not eof(infile) do  
  
        begin  
  
            i := i + 1;  
  
            readln(infile, word[i])  
  
        end;  
  
    close(infile);  
  
    NoOfWord := i;  
  
    randomize;  
  
    Secret := Word[random(NoOfWord)+1]
```

end;

procedure FormQues;

begin

Solution := '';

for i := 1 to length(Secret) do

Solution := Solution + '*';

Chance := 2 * length(Secret)

end;

procedure LetterGuess;

begin

textcolor(12);

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);

```

    Chance := Chance - 1;

    writeln('Chane left = ', Chance)

end;


procedure WinCheck;

begin

    Done := (Solution = Secret) or (Chance = 0)

end;


Procedure WordGuessing;

begin

    textcolor(13);

    repeat

        SelectSecret(Secret);

        FormQues;

        Done := false;

        writeln('The secret word to be guessed is ',Solution);

        writeln('Chance remaning = ',Chance) ;

        writeln;

```

```
repeat
    LetterGuess;
    WinCheck
until Done = true;

if Solution = Secret then
    begin
        writeln('You win!!');
        score[user_index] := score[user_index] +1;
    end
else
    begin
        writeln('You lose!!');
        writeln('The secret is ',secret);
    end;

writeln;

writeln('Continue?(Y/N)');

readln(response);

until (Response='n') or (Response='N');

readln
```

end;

{ Players' account related }

{ Ref:

<http://computer-programming-forum.com/29-pascal/7af4f3f05f738777.htm> }

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

var

S : string;

C : Char;

begin

S := '';

repeat

C := ReadKey;

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

begin

S := S + C;

write('*');

```
        end

    else if C = #8 then

        begin

            S[0] := Chr(Length(S) - 1);

            GotoXY(WhereX - 1, WhereY);

            write(' ');

            GotoXY(WhereX - 1, WhereY);

        end;

    until (C = #10) or (C = #13);

    GetPWord := S;

    writeLn;

end;
```

```
procedure read_users_info;
```

```
var
```

```
    i : integer;
```

```
    f : text;
```

```

begin

    assign(f, 'users.txt');

    reset(f);

    i := 0;

    while not eof(f) do

        begin

            i := i + 1;

            readln(f, userid[i], userpw[i], name[i], gender[i],
score[i]);

            end;

            num_user := i;

            close(f)

        end;

    procedure store_users_info;

    var    i : integer;

            Doc : text;

    begin

        assign(Doc, 'users.txt');

```



```
rewrite(Doc);

for i := 1 to num_user do

    begin

        writeln(Doc, userid[i], userpw[i], name[i], gender[i],
score[i]);

        end;

    close(Doc)

end;
```

```
procedure create_ac;
```

```
var
```

```
    loginid, username, password1, password2 : string;
```

```
    gender_in : char;
```

```
    id_OK, pw_OK : boolean;
```

```
    i : integer;
```

```
begin
```

```
    clrscr;
```

```
    writeln;
```

```
    textcolor(Yellow);
```

```

        writeln('          Your Name with more than 25 characters
will be truncated.');
```

```

        writeln('          Your Name cannot be changed after
created.');
```

```

        textcolor(LightGray);

        write('          Please enter your Name (at most 25 char) :
');

        readln(username);

        repeat

            writeln;

            write('          Please enter your Gender (M or F) : ');

            readln(gender_in)

        until gender_in in ['M','F','m','f'];

        id_OK := false;

        repeat

            clrscr;

            writeln;

            writeln('          Welcome ', username, '!');
```

```

writeln;

write('          Please enter your LoginID (4 char) : ');

readln(loginid);

if length(loginid) <> 4 then

    begin

        writeln;

        writeln('          The length of login must be 4
char!');

        write('          Press <Enter> to retry. ');

        readln

    end

else

    begin

        id_OK := true;

        for i := 1 to num_user do

            if loginid = userid[i] then

                begin

                    writeln;

```

```

        writeln('        This LoginID has been
used!');

        write('        Press <Enter> to enter again.
');

        readln;

        id_OK := false

    end

end

until id_OK;

pw_OK := false;

repeat

    clrscr;

    writeln;

    writeln('        Welcome ', username, '!');

    writeln;

    writeln('        Your LoginID is ', loginid);

    writeln;

    write('        Please enter your password (4 char) : ');

```

```

password1 := GetPword;

if length(password1) <> 4 then

    begin

        writeln;

        writeln('          The length of password must be 4
char!');

        write('          Press <Enter> to retry. ');

        readln

    end

    else

        begin

            writeln;

            write('          Please enter your password
again      : ');

            password2 := GetPword;

            if password1 <> password2 then

                begin

                    writeln;

```

```

        writeln('          The passwords do not
match!');

        write('          Press <Enter> to retry. ');

        readln

    end

else

    begin

        num_user := num_user + 1;

        username := username + '

';

        username := copy(username, 1, 25);

        userid[num_user] := loginid;

        userpw[num_user] := password1;

        name[num_user] := username;

        gender[num_user] := gender_in;

        score[num_user] := 0;

        store_users_info;

```

```

        pw_OK := true;

        writeln;

        writeln('        Your account is created
successfully.');
```

```

        write('        Press <Enter> to return. ');

        readln

    end

end

until pw_OK

end;
```

```

procedure display_user_score(user_index : integer);

begin

    textcolor(10);

    clrscr;

    writeln;

    writeln('        Your results');
```

```

writeln;

writeln('
***** ');

writeln;

writeln('          Your UserID          : ',
userid[user_index]);

writeln;

writeln('          Your name              : ',
name[user_index]);

writeln;

writeln('          Your total score        : ',
score[user_index]);

writeln('
***** ');

writeln;

write('          Press <Enter> to return. ');

readln

end;

```



```

procedure change_password(user_index : integer);

var

    oldpass, newpass1, newpass2 : string;

    pwchanged : boolean;

begin

    pwchanged := false;

    repeat

        clrscr;

        writeln;

        write('          Please enter your old

password          : ');

        oldpass := GetPword;

        if oldpass <> userpw[user_index] then

            begin

                writeln;

                writeln('          Incorrect old password!');

                write('          Press <Enter> to try again. ');

                readln

            end

```

```

else
    begin
        writeln;

        write('          Please enter your new password (4
char) : ');

        newpass1 := GetPword;

        if length(newpass1) <> 4 then
            begin
                writeln;

                writeln('          The length of password must
be 4!');

                write('          Press <Enter> to retry. ');

                readln

            end
        else
            begin
                writeln;

                write('          Please enter your new
password one more time      : ');

```

```

newpass2 := GetPword;

if newpass1 <> newpass2 then

    begin

        writeln;

        writeln('          The new passwords do

not match!');

        write('          Press <Enter> to retry. ');

        readln

    end

else

    begin

        userpw[user_index] := newpass1;

        store_users_info;

        pwchanged := true;

        writeln;

        writeln('          Password has been

changed.');
```

```

        write('          Press <Enter> to return. ');

        readln

```

```

        end
    end
end
until pwchanged
end;

```

```

Procedure main_menu(user_index : integer);

```

```

begin

```

```

repeat

```

```

    textcolor(Yellow);

```

```

    clrscr;

```

```

    writeln;

```

```

    writeln('                                Welcome to Puzzle

```

```

Games ');

```

```

    writeln('                                Please choose:');

```

```

    writeln;

```

```

    writeln('

```

```

    ~~~~~~');

```

```

    writeln;

```

```

writeln('                                1.MasterMind');
writeln;

writeln('                                2.MathGame');
writeln;

writeln('                                3.WordGuessing');
writeln;

writeln('                                4.CheckScore');
writeln;

writeln('                                5.ChangePassword');
writeln;

writeln('                                6.Exit');
writeln;

writeln('
~~~~~');

writeln;

write('                                What is your choice: ');

readln(Choice);

writeln;

case choice of

```

1 : Mastermind;

2 : MathGame;

3 : WordGuessing;

4 : display_user_score(user_index);

5 : change_password(user_index);

end;

until choice = 6;

end;

procedure login(var user_index : integer);

var

loginid, password : string;

found : boolean;

i : integer;

begin

textcolor(12);

clrscr;

```

writeln;

writeln;

writeln('                                Welcome to

Puzzle Game                                ');

writeln;

writeln('

*****

');

writeln('                                *

*                                ');

writeln('                                *                                LOGIN

*                                ');

writeln('                                *

*                                ');

writeln('

*****

');

writeln;

write('                                LoginID: ');

```

```

readln(loginid);

writeln;

write('                Password : ');

password := GetPword;

writeln;

writeln;

found := false;

i := 0;

while (i < num_user) and (not found) do
    begin
        i := i + 1;

        if (loginid = userid[i]) and (password = userpw[i]) then
            begin
                found := true;

                user_index := i
            end
        end;

    if not found then
        begin

```



```

        user_index := 0;

        textcolor(9);

        writeln(':20,'> > > Invalid UserID or Password!');

        write(':20,'> > > Press <Enter> to return.');
```

readln

end

else

 main_menu(user_index)

end;


```

procedure first_page;

begin

    clrscr;

    writeln;

    writeln('                                Welcome to Puzzle

game ');

    writeln('                                Please choose:');
```

```

writeln;

writeln('
=====');

writeln;

writeln('                1. Login');

writeln;

writeln('                2. Create account');

writeln;

writeln('
=====');

writeln;

write('                Enter your choice: ');

readln(Choice);

writeln;

case choice of

    1 : login(user_index);

    2 : create_ac;

end;

end;

```

```
begin
read_users_info;

  repeat

    first_page;

  until false;

  readln
end.
```