Department of
Electronic and Information Engineering

ENG2002 COMPUTER PROGRAMMING

Mini-Project Final Report

Project 2: Lucky888

Lecturer: LEUNG Hung Fat Frank

Tutor: Dr Paul Lai

Group: 56

Group member: ZHANG Kunyuan 13103442D

YANG Tianyu 13102841D

Date: 2015/4/18

Content

Notes: Implement of the class is from Page 7-Page 101.

Abstract

Requirement:

Because we are **Group 56**, we need to follow the **Rule 1)**;

Rule 1) In each turn, a hand of five poker cards will be distributed. The player is allowed to swap at most THREE cards for one time. Whenever a card is selected by clicking it, the card turns facedown. The player can unselect the card by clicking the card once more. After the swapping has been done, the player should click the "EXTRA" button to get one more card for the final hand

Objective:

The objective of this project is to complete a card game named Lucky888 using C++/CLI language and Visual Studio as the programming platform. This game is a gambling game letting users gamble using cards. The player would win in some situations and win the corresponding scores. There are several basic requirements and rules for the game. Besides, we also add some interesting functions to make the game more interesting and complicated. We could let the user seeing the history score list of this game. During playing the game, there is a car running across the screen to make it more interesting. The whole game is based on windows forms application, and could act fluently as a full game program. The report is followed with the introduction, methodology, rules, result, conclusion, further development and user menu.

1. Introduction

1.1 Introduction

The main program consists of many windows forms, the menu form, the rules form, the game form, the new user form, the do you want form, the new game form. There forms make the game interesting and easy to play.

First, if it is the first time for the user to play the game, the user would create the new username and password by himself. Then, a txt file will be created. The name of the file is the user's username, the content is the password and the score. The initial score is 100 for each newcomer. The user could click" RANK" to see the history score list of the previous players. The user could also login with his information. The user clicks" START" to start the game. If the personal information is the same to the information stored in the file, the game form would jump out.

Second, the user would enter the score he wants to pay for this round. The score he want to play should be greater or equal to 20. After checking "START", he would receive 5 cards randomly. According to the situation, he could choose 0, 1, 2 or 3 cards to swap. The substitute cards are different from the previous cards. And then, the user should click" EXTRA" to get a new card which is different from the previous cards. After this, the user would like to click "PLAY" to make the judgment of the score. According to different situations, the user could get different scores proportional to the scores he paid. If he get some scores, he could choose to play another game called" guessing the number ". He would guess the card's number to be showed. He could guess the number is big or small. If he guesses rightly, he could get double score of this round. If he guesses wrong, he would lose all the score he win this round.

Third, the user could play the game many times if he has enough score to pay. If he get bored, he would click" save and quit" to quit the game. His score will be stores in his txt file. The score could also be stored in the history score list if it is in the top 10 order. Maybe his score is 0, then the form will be closed automatically. His file would also store the 0 score.

The historical score list would still store the previous score if he has been on the list. Because we think if he has gotten a top 10 score, his name would be remembered.

1.2 Advanced features and extra-added ones:

Advanced features:

(1) The player is allowed to pay more than 20 credits for a new set of cards. The credits won will then be proportional to the credits paid.
(2) Introduce a joker card as the wildcard that can represent any card.
(3) If the player wins some credits in one turn, he/she is allowed to use the credits won to play a Double-or-None game, say, guessing big or small for an upcoming card. If the player wins this game, the credits would be doubled; otherwise, the credits won would be lost.

Extra-added features:

(1) Add a new function by using fstream to display the rank list of this game. This rank list will display Top 10 player of this game and list their name and credit.
(2) Player can use a special prop to change all five initial cards by paying another 40 credits.
(3) Add the animation by using the timer. A small car will run cross the screen, which will make player more exciting about the game.

2. <u>Methodology</u>

2.1 Team Work Distribution

ZHANG Kunyuan	**YANG Tianyu**
1. Design the logic	**1. Design the logic**

2. login system **3. GUI design** **4. Write Report**	**2. Judging system** **3. GUI design** **4. Write Report**

2.2 Work Schedule

Week No.	Tasks
Week7	Discuss about the mini-project Analysis the project requirements Make a Schedule for programming
Week 8	Think up ideas about new features Confirm the class we need to compile Think up the logic of judgement
Week 9	Finish implementation of the class Think up ideas about GUI layout design Build up and test the static library in order to avoid compiling error
Week 10	Write interim report of mini-project Do the GUI designing and compile and improve the implementation of the class
Week 11	Finish the basic requirement of the game Finish compiling of advanced features
Week 12	Finish compiling of new features Debug the program and improve the functions we used to simplify the program
Week 13	Beautify the GUI layout and pictures Write mini-project Report

2.3 Structure of the Program Developed

2.3.1 The Specification of the Class, Member Variables and Member Functions created

This is the class header file in native C++ library project. The five classes were built in five libraries to implement the whole project. The detailed functions included in the classes are to be discussed in the latter part.

Standard libraries that we need to include:

```
#include<iostream>//for basic input and output
#include<string.h>//for basic string operation
#include<fstream>//for file in and out
#include<stdlib.h>//for atoi or itoa use
using namespace std;
```

1. **Class judge**

Description: This class is used for calculating user's gained credit after playing the game each time. When inputting six integers which represent the value of six cards, this class will use a public function to judge the final gained credit for players. This function will return an integer which represents the gained credit of this game.

```
class judge
{
public:
 void swap(int array1[],int i,int j);//swap the order
 void InsertSort(int array1[],int n);//change the order from the smallest to the biggest
 int Countsuit(int n1,int n2,int n3,int n4,int n5,int n6);//count the largest number of
suit in six cards
```

```
 int judge1(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 1 for six cards
 int judge2(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 2 for six cards
 int judge3(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 3 for six cards
 int judge4(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 4 for six cards
 int judge5(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 5 for six cards
 int judge6(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 6 for six cards
 int judge7(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 7 for six cards
 int judge8(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 8 for six cards
 int judge9(int n1,int n2,int n3,int n4,int n5,int n6);//Judge Rule 9 for six cards
 int Ccredit(int n1,int n2,int n3,int n4,int n5,int n6);//get the credit of the six cards
 int joker(int n1,int n2,int n3,int n4,int n5,int n6);//judge if there is a joker in the
six cards
 int Fcredit(int n1,int n2,int n3,int n4,int n5,int n6);//Judge the final credit for user
private:c
 int n1;
 int n2;
 int n3;
 int n4;
 int n5;
 int n6;
};
```

<u>CLASS IMPLEMENTATION</u>

```
#include"123.h"
```

`void judge::swap(int array1[],int i,int j)`
`{`
`    int tmp=array1[i];`
`    array1[i]=array1[j];`
`    array1[j]=tmp;`
	1. The class is set to calculate the credit the player gain for each game.
	2. It includes three member functions to realize some of

```
}
```

```
void judge::InsertSort(int array1[],int n)
{
    for(int i=1;i<n;i++)
    {
        for(int j=i;j>0;j--)
        {
            if(array1[j]<array1[j-1])
                swap(array1,j,j-1);  //Swap the order if the latter one is smaller than the former one
            else
                break;
        }
    }
}
```

```
int judge::Countsuit(int n1,int n2,int n3,int n4,int n5,int n6)
{
     int credit;
     int card[6]={0,0,0,0,0,0};
     card[0]=n1;
     card[1]=n2;
     card[2]=n3;
     card[3]=n4;
     card[4]=n5;
     card[5]=n6;
     int suit[4]={0,0,0,0};
     for(int i=0;i<6;i++)
     {
         if ((card[i]>=0)&&(card[i]<=12))
```

the functions which will be used in other member function.

3. It includes nine member function to calculate the credit for each case:

(1) Royal Flush

(2) Straight Flush

(3) Four of a Kind

(4) Full House

(5) Flush

(6) Straight

(7) Three of a Kind

(8) Two Pairs

(9) One Pair

4. It includes a function to calculate the credit for the normal case(no joker credit)

5. It includes another function to calculate the credit when there is a joker.(If there is no joker, this function will return 0)

```
        {
        suit[0]++;   //Calculate the number of clubs in six cards
        }
        if ((card[i]>=13)&&(card[i]<=25))
        {
        suit[1]++;   //Calculate the number of diamonds in six cards
        }
        if ((card[i]>=26)&&(card[i]<=38))
        {
        suit[2]++;   //Calculate the number of hearts in six cards
        }
        if ((card[i]>=39)&&(card[i]<=51))
        {
        suit[3]++;   //Calculate the number of spades in six cards
        }
    }
    InsertSort(suit,4);

    return suit[3];
}
```

```
int judge::judge1(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit;
    int card[6]={0,0,0,0,0,0};
    card[0]=n1;
    card[1]=n2;
    card[2]=n3;
    card[3]=n4;
    card[4]=n5;
    card[5]=n6;


    int array1[5]={0,0,0,0,0};
    array1[0]=card[0];
    array1[1]=card[1];
    array1[2]=card[2];
    array1[3]=card[3];
    array1[4]=card[4];
    InsertSort(array1,5);//change the order from small to big
    //Judge each five cards in six cards(six times) if there has royal flush
in six cards
    if
((array1[0]==8&&array1[4]==12)||(array1[0]==21&&array1[4]==25)||(array1[0
]==34&&array1[4]==38)||(array1[0]==47&&array1[4]==51))
    {credit=10000;
    }else
    {


    int array2[5]={0,0,0,0,0};
    array2[0]=card[0];
    array2[1]=card[1];
    array2[2]=card[2];
    array2[3]=card[3];
    array2[4]=card[5];
    InsertSort(array2,5);
    if
((array2[0]==8&&array2[4]==12)||(array2[0]==21&&array2[4]==25)||(array2[0
]==34&&array2[4]==38)||(array2[0]==47&&array2[4]==51))
```

```
    {credit=10000;
    }else
    {


    int array3[5]={0,0,0,0,0};
    array3[0]=card[0];
    array3[1]=card[1];
    array3[2]=card[2];
    array3[3]=card[4];
    array3[4]=card[5];
    InsertSort(array3,5);
    if
((array3[0]==8&&array3[4]==12)||(array3[0]==21&&array3[4]==25)||(array3[0
]==34&&array3[4]==38)||(array3[0]==47&&array3[4]==51))
    {credit=10000;
    }
    else{


    int array4[5]={0,0,0,0,0};
    array4[0]=card[0];
    array4[1]=card[1];
    array4[2]=card[3];
    array4[3]=card[4];
    array4[4]=card[5];
    InsertSort(array4,5);
    if
((array4[0]==8&&array4[4]==12)||(array4[0]==21&&array4[4]==25)||(array4[0
]==34&&array4[4]==38)||(array4[0]==47&&array4[4]==51))
    {credit=10000;
    }else
    {
    int array5[5]={0,0,0,0,0};
    array5[0]=card[0];
    array5[1]=card[2];
    array5[2]=card[3];
    array5[3]=card[4];
    array5[4]=card[5];
    InsertSort(array5,5);
    if
((array5[0]==8&&array5[4]==12)||(array5[0]==21&&array5[4]==25)||(array5[0
]==34&&array5[4]==38)||(array5[0]==47&&array5[4]==51))
    {credit=10000;
    }else
    {
    int array6[5]={0,0,0,0,0};
    array6[0]=card[1];
    array6[1]=card[2];
    array6[2]=card[3];
    array6[3]=card[4];
    array6[4]=card[5];
    InsertSort(array6,5);
    if
((array6[0]==8&&array6[4]==12)||(array6[0]==21&&array6[4]==25)||(array6[0
]==34&&array6[4]==38)||(array6[0]==47&&array6[4]==51))
    {credit=10000;
    }else{credit=0;}
    }}}}}
    return credit;
```

```
}
```

```
int judge::judge2(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit;
    int card[6]={0,0,0,0,0,0};
    card[0]=n1;
    card[1]=n2;
    card[2]=n3;
    card[3]=n4;
    card[4]=n5;
    card[5]=n6;

    int array1[5]={0,0,0,0,0};
    array1[0]=card[0];
    array1[1]=card[1];
    array1[2]=card[2];
    array1[3]=card[3];
    array1[4]=card[4];
    InsertSort(array1,5);
    int a=array1[0]%13;
    int b=array1[4]&13;
    //Judge each five cards in six cards(six times) if there has straight flush
in six cards
    if((array1[0]+4==array1[4])&&(a<=8)&&(b>=4))
    {credit=4000;
    }else
    {
```

```
int array2[5]={0,0,0,0,0};
array2[0]=card[0];
array2[1]=card[1];
array2[2]=card[2];
array2[3]=card[3];
array2[4]=card[5];
InsertSort(array2,5);
int a=array2[0]%13;
int b=array2[4]&13;

if((array2[0]+4==array2[4])&&(a<=8)&&(b>=4))
{credit=4000;
}else
{

int array3[5]={0,0,0,0,0};
array3[0]=card[0];
array3[1]=card[1];
array3[2]=card[2];
array3[3]=card[4];
array3[4]=card[5];
InsertSort(array3,5);
int a=array3[0]%13;
int b=array3[4]&13;

if((array3[0]+4==array3[4])&&(a<=8)&&(b>=4))
{credit=4000;
}
else{
```

```
int array4[5]={0,0,0,0,0};
array4[0]=card[0];
array4[1]=card[1];
array4[2]=card[3];
array4[3]=card[4];
array4[4]=card[5];
InsertSort(array4,5);
int a=array4[0]%13;
int b=array4[4]&13;

if((array4[0]+4==array4[4])&&(a<=8)&&(b>=4))
{credit=4000;
}else
{
int array5[5]={0,0,0,0,0};
array5[0]=card[0];
array5[1]=card[2];
array5[2]=card[3];
array5[3]=card[4];
array5[4]=card[5];
InsertSort(array5,5);
int a=array5[0]%13;
int b=array5[4]&13;

if((array5[0]+4==array5[4])&&(a<=8)&&(b>=4))
{credit=4000;
}else
{
    int array6[5]={0,0,0,0,0};
    array6[0]=card[1];
    array6[1]=card[2];
    array6[2]=card[3];
    array6[3]=card[4];
    array6[4]=card[5];
    InsertSort(array6,5);
    int a=array6[0]%13;
    int b=array6[4]&13;

    if((array6[0]+4==array6[4])&&(a<=8)&&(b>=4))
    {credit=4000;
    }else{credit=0;}
    }}}}
    return credit;
}
```

```
int judge::judge3(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit;
    int card[6]={0,0,0,0,0,0};
    card[0]=n1;
    card[1]=n2;
    card[2]=n3;
    card[3]=n4;
    card[4]=n5;
    card[5]=n6;

    for(int i=0;i<6;i++)
    {
        card[i]=card[i]%13;
    }
    int array1[5]={0,0,0,0,0};
    array1[0]=card[0];
    array1[1]=card[1];
    array1[2]=card[2];
    array1[3]=card[3];
    array1[4]=card[4];
    InsertSort(array1,5);
    //Judge each five cards in six cards(six times) if there has four of a
kinds in six cards
    if
((array1[0]==array1[1]&&array1[0]==array1[3])||(array1[4]==array1[3]&&arr
ay1[4]==array1[1]))//If the smallest one is equal to the biggest one, this
five cards is equal
        credit=800;
    else
    {
    int array2[5]={0,0,0,0,0};
    array2[0]=card[0];
    array2[1]=card[1];
    array2[2]=card[2];
    array2[3]=card[3];
    array2[4]=card[5];
    InsertSort(array2,5);

    if
((array2[0]==array2[1]&&array2[0]==array2[3])||(array2[4]==array2[3]&&arr
ay2[4]==array2[1]))
        credit=800;
    else
    {
    int array3[5]={0,0,0,0,0};
    array3[0]=card[0];
    array3[1]=card[1];
    array3[2]=card[2];
    array3[3]=card[4];
    array3[4]=card[5];
    InsertSort(array3,5);

    if
((array3[0]==array3[1]&&array3[0]==array3[3])||(array3[4]==array3[3]&&arr
ay3[4]==array3[1]))
        credit=800;
    else
    {
    int array4[5]={0,0,0,0,0};
    array4[0]=card[0];
    array4[1]=card[1];
    array4[2]=card[3];
    array4[3]=card[4];
    array4[4]=card[5];
    InsertSort(array4,5);

    if
((array4[0]==array4[1]&&array4[0]==array4[3])||(array4[4]==array4[3]&&arr
ay4[4]==array4[1]))
```

```
        credit=800;
    else
    {
    int array5[5]={0,0,0,0,0};
    array5[0]=card[0];
    array5[1]=card[2];
    array5[2]=card[3];
    array5[3]=card[4];
    array5[4]=card[5];
    InsertSort(array5,5);

    if
((array5[0]==array5[1]&&array5[0]==array5[3])||(array5[4]==array5[3]&&arr
ay5[4]==array5[1]))
        credit=800;
    else
    {
    int array6[5]={0,0,0,0,0};
    array6[0]=card[1];
    array6[1]=card[2];
    array6[2]=card[3];
    array6[3]=card[4];
    array6[4]=card[5];
    InsertSort(array6,5);

    if
((array6[0]==array6[1]&&array6[0]==array6[3])||(array6[4]==array6[3]&&arr
ay6[4]==array6[1]))
        credit=800;
    else
    {
    credit=0;
    }
    }
    }
    }
    }
    }
    return credit;
}
```

```
int judge::judge4(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit;
    int card1[6]={0,0,0,0,0,0};
    card1[0]=n1;
    card1[1]=n2;
    card1[2]=n3;
    card1[3]=n4;
    card1[4]=n5;
    card1[5]=n6;

    int card[6];
    for(int i=0;i<6;i++)
    {
        card[i]=card1[i]%13;
    }
    int array1[5]={0,0,0,0,0};
    array1[0]=card[0];
    array1[1]=card[1];
    array1[2]=card[2];
    array1[3]=card[3];
    array1[4]=card[4];
    InsertSort(array1,5);
    //Judge each five cards in six cards(six times) if there has full house
in six cards
    if
((array1[0]==array1[1]&&array1[2]==array1[3]&&array1[2]==array1[4])||(arr
ay1[4]==array1[3]&&array1[0]==array1[1]&&array1[2]==array1[1]))
        credit=200;
    else
    {
    int array2[5]={0,0,0,0,0};
    array2[0]=card[0];
    array2[1]=card[1];
    array2[2]=card[2];
    array2[3]=card[3];
    array2[4]=card[5];
    InsertSort(array2,5);

    if
((array2[0]==array2[1]&&array2[2]==array2[3]&&array2[2]==array2[4])||(arr
ay2[4]==array2[3]&&array2[0]==array2[1]&&array2[2]==array2[1]))
        credit=200;
    else
    {
```

```
    int array3[5]={0,0,0,0,0};
    array3[0]=card[0];
    array3[1]=card[1];
    array3[2]=card[2];
    array3[3]=card[4];
    array3[4]=card[5];
    InsertSort(array3,5);

    if
((array3[0]==array3[1]&&array3[2]==array3[3]&&array3[2]==array3[4])||(arr
ay3[4]==array3[3]&&array3[0]==array3[1]&&array3[2]==array3[1]))
        credit=200;
    else
    {
    int array4[5]={0,0,0,0,0};
    array4[0]=card[0];
    array4[1]=card[1];
    array4[2]=card[3];
    array4[3]=card[4];
    array4[4]=card[5];
    InsertSort(array4,5);

    if
((array4[0]==array4[1]&&array4[2]==array4[3]&&array4[2]==array4[4])||(arr
ay4[4]==array4[3]&&array4[0]==array4[1]&&array4[2]==array4[1]))
        credit=200;
    else
    {
    int array5[5]={0,0,0,0,0};
    array5[0]=card[0];

    array5[1]=card[2];

    array5[2]=card[3];

    array5[3]=card[4];

    array5[4]=card[5];

    InsertSort(array5,5);


    if
((array5[0]==array5[1]&&array5[2]==array5[3]&&array5[2]==array5[4])||(arr
ay5[4]==array5[3]&&array5[0]==array5[1]&&array5[2]==array5[1]))

        credit=200;

    else

    {

    int array6[5]={0,0,0,0,0};

    array6[0]=card[1];

    array6[1]=card[2];

    array6[2]=card[3];

    array6[3]=card[4];

    array6[4]=card[5];

    InsertSort(array6,5);


    if
((array6[0]==array6[1]&&array6[2]==array6[3]&&array6[2]==array6[4])||(arr
ay6[4]==array6[3]&&array6[0]==array6[1]&&array6[2]==array6[1]))

        credit=200;

    else

    {

    credit=0;

    }
    }
    }
    }
    }
    }
    return credit;
}
```

```
int judge::judge5(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit;
    int card[6]={0,0,0,0,0,0};
    card[0]=n1;
    card[1]=n2;
    card[2]=n3;
    card[3]=n4;
    card[4]=n5;
    card[5]=n6;
    //Judge each five cards in six cards(six times) if there has flush in six cards
    if (Countsuit(card[0],card[1],card[2],card[3],card[4],card[5])>=5)
        credit=150;
    else
        credit=0;
    return credit;
}
```

```
int judge::judge6(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit;

    int card[6]={0,0,0,0,0,0};

    card[0]=n1;

    card[1]=n2;

    card[2]=n3;

    card[3]=n4;

    card[4]=n5;

    card[5]=n6;


    int tmp[6];

    for(int i=0;i<6;i++)

    {

        tmp[i]=card[i]%13;

    }




    int array1[5]={0,0,0,0,0};

    array1[0]=tmp[0];

    array1[1]=tmp[1];

    array1[2]=tmp[2];

    array1[3]=tmp[3];

    array1[4]=tmp[4];

    InsertSort(array1,5);

    //Judge each five cards in six cards(six times) if there has straight in
six cards

    if
(array1[1]==array1[0]+1&&array1[2]==array1[1]+1&&array1[3]==array1[2]+1&&
array1[4]==array1[3]+1)

    {
        credit=100;}
    else{
    int array2[5]={0,0,0,0,0};
    array2[0]=tmp[0];
    array2[1]=tmp[1];
    array2[2]=tmp[2];
    array2[3]=tmp[3];
    array2[4]=tmp[5];
    InsertSort(array2,5);

    if
(array2[1]==array2[0]+1&&array2[2]==array2[1]+1&&array2[3]==array2[2]+1&&
array2[4]==array1[3]+1)
    {
        credit=100;}
    else{

    int array3[5]={0,0,0,0,0};
    array3[0]=tmp[0];
    array3[1]=tmp[1];
    array3[2]=tmp[2];
    array3[3]=tmp[4];
    array3[4]=tmp[5];
    InsertSort(array3,5);


    if
(array3[1]==array3[0]+1&&array3[2]==array3[1]+1&&array3[3]==array3[2]+1&&
array3[4]==array3[3]+1)
```

```
    {
        credit=100;}else{



    int array4[5]={0,0,0,0,0};
    array4[0]=tmp[0];
    array4[1]=tmp[1];
    array4[2]=tmp[3];
    array4[3]=tmp[4];
    array4[4]=tmp[5];
    InsertSort(array4,5);

    if
(array4[1]==array4[0]+1&&array4[2]==array4[1]+1&&array4[3]==array4[2]+1&&
array4[4]==array4[3]+1)
    {
        credit=100;}else{
    int array5[5]={0,0,0,0,0};
    array5[0]=tmp[0];
    array5[1]=tmp[2];
    array5[2]=tmp[3];
    array5[3]=tmp[4];
    array5[4]=tmp[5];
    InsertSort(array5,5);
    if
(array5[1]==array5[0]+1&&array5[2]==array5[1]+1&&array5[3]==array5[2]+1&&
array5[4]==array5[3]+1)
    {
        credit=100;}else{
```

```
    int array6[5]={0,0,0,0,0};
    array6[0]=tmp[1];
    array6[1]=tmp[2];
    array6[2]=tmp[3];
    array6[3]=tmp[4];
    array6[4]=tmp[5];
    InsertSort(array6,5);


    if
(array6[1]==array6[0]+1&&array6[2]==array6[1]+1&&array6[3]==array6[2]+1&&
array6[4]==array6[3]+1)
    {
        credit=100;}
    else{


            credit=0;
        }
        }
        }
    }
    }}
    return credit;
}
```

```
int judge::judge7(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit;
    int card[6]={0,0,0,0,0,0};
    card[0]=n1;
    card[1]=n2;
    card[2]=n3;
    card[3]=n4;
    card[4]=n5;
    card[5]=n6;
    int tmp[6];
    for(int i=0;i<6;i++)
    {
        tmp[i]=card[i]%13;
    }
    int array1[5]={0,0,0,0,0};
    array1[0]=tmp[0];
    array1[1]=tmp[1];
    array1[2]=tmp[2];
    array1[3]=tmp[3];
    array1[4]=tmp[4];
    InsertSort(array1,5);
//Judge each five cards in six cards(six times) if there has three of a kind
in six cards
    if
((array1[0]==array1[2])||(array1[1]==array1[3])||(array1[2]==array1[4]))
    {
        credit=60;}
    else
    {
    int array2[5]={0,0,0,0,0};
    array2[0]=tmp[0];
    array2[1]=tmp[1];
    array2[2]=tmp[2];
    array2[3]=tmp[3];

    array2[4]=tmp[5];

    InsertSort(array2,5);


    if
((array2[0]==array2[2])||(array2[1]==array2[3])||(array2[2]==array2[4]))

    {

        credit=60;}

    else

    {

    int array3[5]={0,0,0,0,0};

    array3[0]=tmp[0];

    array3[1]=tmp[1];

    array3[2]=tmp[2];

    array3[3]=tmp[4];

    array3[4]=tmp[5];

    InsertSort(array3,5);


    if
((array3[0]==array3[2])||(array3[1]==array3[3])||(array3[2]==array3[4]))

    {

        credit=60;}

    else

    {

    int array4[5]={0,0,0,0,0};

    array4[0]=tmp[0];

    array4[1]=tmp[1];

    array4[2]=tmp[3];

    array4[3]=tmp[4];
    array4[4]=tmp[5];
    InsertSort(array4,5);


    if
((array4[0]==array4[2])||(array4[1]==array4[3])||(array4[2]==array4[4]))
    {
        credit=60;}
    else
    {
    int array5[5]={0,0,0,0,0};
    array5[0]=tmp[0];
    array5[1]=tmp[2];
    array5[2]=tmp[3];
    array5[3]=tmp[4];
    array5[4]=tmp[5];
    InsertSort(array5,5);


    if
((array5[0]==array5[2])||(array5[1]==array5[3])||(array5[2]==array1[4]))
    {
        credit=60;}else
        {
        int array6[5]={0,0,0,0,0};
    array6[0]=tmp[1];
    array6[1]=tmp[2];
    array6[2]=tmp[3];
    array6[3]=tmp[4];
    array6[4]=tmp[5];
    InsertSort(array6,5);
    if
((array6[0]==array6[2])||(array6[1]==array6[3])||(array6[2]==array6[4]))
    {
        credit=60;}else{
        credit=0;
        }}
        }
    }
    }
    }
    return credit;
}
```

```
int judge::judge8(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit;
    int card[6]={0,0,0,0,0,0};
    card[0]=n1;
    card[1]=n2;
    card[2]=n3;
    card[3]=n4;
    card[4]=n5;
    card[5]=n6;
    int tmp[6];
    for(int i=0;i<6;i++)
    {
        tmp[i]=card[i]%13;
    }
    int array1[5]={0,0,0,0,0};
    array1[0]=tmp[0];
    array1[1]=tmp[1];
    array1[2]=tmp[2];
    array1[3]=tmp[3];
    array1[4]=tmp[4];
    InsertSort(array1,5);
    //Judge each five cards in six cards(six times) if there has two pairs
in six cards
    if(((array1[0]==array1[1])&&(array1[2]==array1[3]))||((array1[0]==arr
ay1[1])&&(array1[3]==array1[4]))||((array1[1]==array1[2])&&(array1[3]==ar
ray1[4])))
    {
        credit=40;}
    else
    {
    int array2[5]={0,0,0,0,0};
    array2[0]=tmp[0];
    array2[1]=tmp[1];
    array2[2]=tmp[2];
    array2[3]=tmp[3];
    array2[4]=tmp[5];
    InsertSort(array2,5);

    if(((array2[0]==array2[1])&&(array2[2]==array2[3]))||((array2[0]==arr
ay2[1])&&(array2[3]==array2[4]))||((array2[1]==array2[2])&&(array2[3]==ar
ray2[4])))
    {
        credit=40;}
    else
    {
    int array3[5]={0,0,0,0,0};
    array3[0]=tmp[0];
    array3[1]=tmp[1];
    array3[2]=tmp[2];
    array3[3]=tmp[4];
    array3[4]=tmp[5];
    InsertSort(array3,5);

    if(((array3[0]==array3[1])&&(array3[2]==array3[3]))||((array3[0]==arr
ay3[1])&&(array3[3]==array3[4]))||((array3[1]==array3[2])&&(array3[3]==ar
ray3[4])))
    {
        credit=40;}
    else
    {
    int array4[5]={0,0,0,0,0};
    array4[0]=tmp[0];
    array4[1]=tmp[1];
    array4[2]=tmp[3];
    array4[3]=tmp[4];
    array4[4]=tmp[5];
    InsertSort(array4,5);

    if(((array4[0]==array4[1])&&(array4[2]==array4[3]))||((array4[0]==arr
ay4[1])&&(array4[3]==array4[4]))||((array4[1]==array4[2])&&(array4[3]==ar
ray4[4])))
    {
        credit=40;}
    else
    {
    int array5[5]={0,0,0,0,0};
    array5[0]=tmp[0];
    array5[1]=tmp[2];
    array5[2]=tmp[3];
    array5[3]=tmp[4];
    array5[4]=tmp[5];
    InsertSort(array5,5);

    if(((array5[0]==array5[1])&&(array5[2]==array5[3]))||((array5[0]==arr
ay5[1])&&(array5[3]==array5[4]))||((array5[1]==array5[2])&&(array5[3]==ar
ray5[4])))
    {
        credit=40;}
    else
        {
        int array6[5]={0,0,0,0,0};
    array6[0]=tmp[1];
    array6[1]=tmp[2];
    array6[2]=tmp[3];
    array6[3]=tmp[4];
    array6[4]=tmp[5];
    InsertSort(array6,5);

    if(((array6[0]==array6[1])&&(array6[2]==array6[3]))||((array6[0]==arr
ay6[1])&&(array6[3]==array6[4]))||((array6[1]==array6[2])&&(array6[3]==ar
ray6[4])))
```

```
    {
        credit=40;}
    else{
        credit=0;
        }}
        }
    }
    }
    }
    return credit;
}
```

```
int judge::judge9(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit=0;
    int card[6]={0,0,0,0,0,0};
    card[0]=n1;
    card[1]=n2;
    card[2]=n3;
    card[3]=n4;
    card[4]=n5;
    card[5]=n6;
    int tmp[6];
    for(int i=0;i<6;i++)
    {
        tmp[i]=card[i]%13;
    }
    int array1[5]={0,0,0,0,0};
    array1[0]=tmp[0];
    array1[1]=tmp[1];
    array1[2]=tmp[2];
    array1[3]=tmp[3];
    array1[4]=tmp[4];
    InsertSort(array1,5);
    //Judge each five cards in six cards(six times) if there has one pair in
six cards
    for(int i=0;i<5;i++)
    {
        for(int j=1;j<5;j++)
        {
            int a=(i+j)%5;
            if (array1[i]==array1[a])
        {credit=20;
        }
        }
    }

    int array2[5]={0,0,0,0,0};
    array2[0]=tmp[0];
    array2[1]=tmp[1];
    array2[2]=tmp[2];
    array2[3]=tmp[3];
    array2[4]=tmp[5];
    InsertSort(array2,5);
    for(int i=0;i<5;i++)
    {
        for(int j=1;j<5;j++)
        {
            int a=(i+j)%5;
            if (array2[i]==array2[a])
        {credit=20;
        }
        }
    }

    int array3[5]={0,0,0,0,0};
    array3[0]=tmp[0];
    array3[1]=tmp[1];
    array3[2]=tmp[2];
    array3[3]=tmp[4];
    array3[4]=tmp[5];
    InsertSort(array3,5);

    for(int i=0;i<5;i++)
    {
        for(int j=1;j<5;j++)
        {
            int a=(i+j)%5;
            if (array3[i]==array3[a])
        {credit=20;
        }
        }
    }
```

```
int array4[5]={0,0,0,0,0};

array4[0]=tmp[0];

array4[1]=tmp[1];

array4[2]=tmp[3];

array4[3]=tmp[4];

array4[4]=tmp[5];

InsertSort(array4,5);


for(int i=0;i<5;i++)

{

    for(int j=1;j<5;j++)

    {

        int a=(i+j)%5;

        if (array4[i]==array4[a])

    {credit=20;

    }

    }

}


int array5[5]={0,0,0,0,0};

array5[0]=tmp[0];

array5[1]=tmp[2];

array5[2]=tmp[3];

array5[3]=tmp[4];

array5[4]=tmp[5];

InsertSort(array1,5);


for(int i=0;i<5;i++)

{
    for(int j=1;j<5;j++)
    {
         int a=(i+j)%5;
         if (array5[i]==array5[a])
    {credit=20;
    }
    }
}



int array6[5]={0,0,0,0,0};
array6[0]=tmp[1];
array6[1]=tmp[2];
array6[2]=tmp[3];
array6[3]=tmp[4];
array6[4]=tmp[5];
InsertSort(array6,5);

for(int i=0;i<5;i++)
{
    for(int j=1;j<5;j++)
    {
         int a=(i+j)%5;
         if (array6[i]==array6[a])
    {credit=20;
    }
    }
}
return credit;


}
```

```
int judge::joker(int n1,int n2,int n3,int n4,int n5,int n6)

{

    int tcredit[6];

    int array1[6];

    array1[0]=n1;

    array1[1]=n2;

    array1[2]=n3;

    array1[3]=n4;

    array1[4]=n5;

    array1[5]=n6;

    for (int i=0;i<6;i++)

    {

        if (array1[i]==52)

        {

            int a=array1[i];

            int b=array1[(i+1)%6];

            int c=array1[(i+2)%6];

            int d=array1[(i+3)%6];

            int e=array1[(i+4)%6];

            int f=array1[(i+5)%6];

            int array2[52];

            for(int j=0;j<52;j++)

            {

                if ((j==b)||(j==c)||(j==d)||(j==e)||(j==f))

                {
```

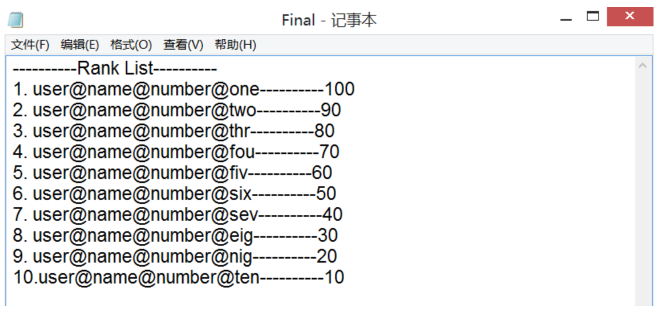
```
                tcredit[i]=0;
                array2[j]=0;}//If there is not a joker in six cards, credit is equal to 0
                else
                {
                a=j;//let the value of the joker card equal to every value from 0-51
                array2[j]=Ccredit(a,b,c,d,e,f);//If there is a joker in six cards, save the credits into an array when joker is equal each of the cards
                }
            }
            InsertSort(array2,52);
            tcredit[i]=array2[51];//Let the credit be the biggest value of 52 values
        }
        else
        {
            tcredit[i]=Ccredit(n1,n2,n3,n4,n5,n6);//Judge
        }
    }
    InsertSort(tcredit,6);//Judge each five cards in six cards(six times) if there is a joker in six cards
    return tcredit[5];
}
```

```
int judge::Ccredit(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int credit[9];
    credit[0]=judge1(n1,n2,n3,n4,n5,n6);
    credit[1]=judge2(n1,n2,n3,n4,n5,n6);
    credit[2]=judge3(n1,n2,n3,n4,n5,n6);
    credit[3]=judge4(n1,n2,n3,n4,n5,n6);
    credit[4]=judge5(n1,n2,n3,n4,n5,n6);
    credit[5]=judge6(n1,n2,n3,n4,n5,n6);
    credit[6]=judge7(n1,n2,n3,n4,n5,n6);
    credit[7]=judge8(n1,n2,n3,n4,n5,n6);
    credit[8]=judge9(n1,n2,n3,n4,n5,n6);
    InsertSort(credit,9);//Judge the credit of nine judge rules and make the
biggest one be the cgained credit
return credit[8];
}
```

```
int judge::Fcredit(int n1,int n2,int n3,int n4,int n5,int n6)
{
    int fcredit;
    if (Ccredit(n1,n2,n3,n4,n5,n6)>=joker(n1,n2,n3,n4,n5,n6))//Judge the
credit between the normal credit and the credit with joker and get the bigger
one
        fcredit=Ccredit(n1,n2,n3,n4,n5,n6);
    else
        fcredit=joker(n1,n2,n3,n4,n5,n6);
    return fcredit;
}
```

2. Class list

Description: This class is to create a rank list which is named ”Final.txt”. When the player finish playing the game, it will save the user’s name and his/her credit into the rank list if his/her credit is of the top ten of all user.



```
----------Rank List----------
1. user@name@number@one----------100
2. user@name@number@two----------90
3. user@name@number@thr----------80
4. user@name@number@fou----------70
5. user@name@number@fiv----------60
6. user@name@number@six----------50
7. user@name@number@sev----------40
8. user@name@number@eig----------30
9. user@name@number@nig----------20
10.user@name@number@ten----------10
```

```
class list
{
public:
 int  convert(char array0[6]);//convert an array of character into an array of integer
 void transfer(char String2[6],int credit);//change the credit(character) into
credit(integer);
 void gg(char name[20],int credit);//Build the rank list for this user(Update the rank
list)
private:
 char buffer[280];
 char whole[260];
 int credit;
 char name[20];
 char array0[6];
};
```

<u>CLASS IMPLEMENTATION</u>

```
#include<iostream>
#include<fstream>
#include<string.h>
using namespace std;
#include"rank.h"C
```

`int  list::convert(char array0[6])`	1. This class includes three

```
{
     int p;
     int array1[6];
     for(int i=0;i<6;i++)
    {
     array1[i]=(int)(array0[i] - '0');//change every digital of the credit into character
    }
     p=100000*array1[0]+10000*array1[1]+1000*array1[2]+100*array1[3]+10*array1[4]+array1[5];//combine every digital into the credit(integer)
     return p;
}
```

```
void list::transfer(char String2[6],int credit)
{
     int String1[6]={0,0,0,0,0,0};
     //get each digital of the credit
     String1[0]=credit/100000;
     String1[1]=(credit-100000*String1[0])/10000;
     String1[2]=(credit-100000*String1[0]-10000*String1[1])/1000;
     String1[3]=(credit-100000*String1[0]-10000*String1[1]-1000*String1[2])/100;
     String1[4]=(credit-100000*String1[0]-10000*String1[1]-1000*String1[2]-100*String1[3])/10;
     String1[5]=(credit-100000*String1[0]-10000*String1[1]-1000*String1[2]-100*String1[3]-10*String1[4])/1;
     for(int i=0;i<6;i++)
     {
          String2[i]=String1[i]+'0';//change it into character
```

member functions in order to build rank list file

(1) Convert an array of character into an array of integer;
(2) Transfer the credit(integer) into an array of character
(3) Build rank list file

2. The function to build rank list file is using the following methods:

(1) There is an original rank list file (blank) in the document.
(2) Pick up 10 names and credits from the original rank list
(3) Judge the new player's credit if he is at the top ten of the rank list
(4) Write the player's name and credit into the final rank list file("Final.txt")
(5) Update new players profile in the original rank list file (This means

```
     }


}
```
	that next time when another player play the game, the original rank list file will be the updated rank list file)

```
void list::gg(char name[20],int credit)


{
char buffer[280];
fstream out;
out.open("Rank List.txt",ios::in);
while(!out.eof())
{
   out.getline(buffer,280,'\n');//Save all content into buffer;
}
out.close();
char whole[260];
strncpy(whole,buffer,260);//Save 260 digit of buffer into whole(10 user's
name and 10 user's credit)
//get each user's name and credit of rank list
char name1[20];
strncpy(name1,whole,20);
char credit1[6];
strncpy(credit1,whole+20,6);
char name2[20];
strncpy(name2,whole+26,20);
char credit2[6];
strncpy(credit2,whole+46,6);
char name3[20];
strncpy(name3,whole+52,20);
```

```
char credit3[6];

strncpy(credit3,whole+72,6);

char name4[20];

strncpy(name4,whole+78,20);

char credit4[6];

strncpy(credit4,whole+98,6);

char name5[20];

strncpy(name5,whole+104,20);

char credit5[6];

strncpy(credit5,whole+124,6);

char name6[20];

strncpy(name6,whole+130,20);

char credit6[6];

strncpy(credit6,whole+150,6);

char name7[20];

strncpy(name7,whole+156,20);

char credit7[6];

strncpy(credit7,whole+176,6);

char name8[20];

strncpy(name8,whole+182,20);

char credit8[6];

strncpy(credit8,whole+202,6);

char name9[20];

strncpy(name9,whole+208,20);

char credit9[6];

strncpy(credit9,whole+228,6);

char name0[20];

strncpy(name0,whole+234,20);

char credit0[6];
```

```
strncpy(credit0,whole+254,6);


char rcredit1[6];

char rcredit2[6];

char rcredit3[6];

char rcredit4[6];

char rcredit5[6];

char rcredit6[6];

char rcredit7[6];

char rcredit8[6];

char rcredit9[6];

char rcredit0[6];

//Build new file named "Final.txt" to build rank list file

ofstream f1("Final.txt");

    f1<<"----------Rank List----------"<<endl;




//Save new player's profile(name,credit)into the rank list if he/she is at
the top ten of the rank list

    // If the player is not at the top ten of the rank list, the original rank
list will not change

if(credit<=convert(credit0))

{

    f1<<"1. ";

    for(int i=0;i<20;i++)

    {f1<<name1[i];}

    f1<<"----------"<<convert(credit1)<<endl;


    f1<<"2. ";
```

```
for(int i=0;i<20;i++)
{f1<<name2[i];}
f1<<"----------"<<convert(credit2)<<endl;

f1<<"3. ";
for(int i=0;i<20;i++)
{f1<<name3[i];}
f1<<"----------"<<convert(credit3)<<endl;

f1<<"4. ";
for(int i=0;i<20;i++)
{f1<<name4[i];}
f1<<"----------"<<convert(credit4)<<endl;

f1<<"5. ";
for(int i=0;i<20;i++)
{f1<<name5[i];}
f1<<"----------"<<convert(credit5)<<endl;

f1<<"6. ";
for(int i=0;i<20;i++)
{f1<<name6[i];}
f1<<"----------"<<convert(credit6)<<endl;

f1<<"7. ";
for(int i=0;i<20;i++)
{f1<<name7[i];}
f1<<"----------"<<convert(credit7)<<endl;
```

```
    f1<<"8. ";

    for(int i=0;i<20;i++)

    {f1<<name8[i];}

    f1<<"----------"<<convert(credit8)<<endl;


    f1<<"9. ";

    for(int i=0;i<20;i++)

    {f1<<name9[i];}

    f1<<"----------"<<convert(credit9)<<endl;


    f1<<"10.";

    for(int i=0;i<20;i++)

    {f1<<name0[i];}

    f1<<"----------"<<convert(credit0)<<endl;


}

else

{

    if((credit<=convert(credit9))&&(credit>convert(credit0)))

    {

        f1<<"1. ";

    for(int i=0;i<20;i++)

    {f1<<name1[i];}

    f1<<"----------"<<convert(credit1)<<endl;


    f1<<"2. ";

    for(int i=0;i<20;i++)

    {f1<<name2[i];}

    f1<<"----------"<<convert(credit2)<<endl;
```

```
f1<<"3. ";
for(int i=0;i<20;i++)
{f1<<name3[i];}
f1<<"----------"<<convert(credit3)<<endl;

f1<<"4. ";
for(int i=0;i<20;i++)
{f1<<name4[i];}
f1<<"----------"<<convert(credit4)<<endl;

f1<<"5. ";
for(int i=0;i<20;i++)
{f1<<name5[i];}
f1<<"----------"<<convert(credit5)<<endl;

f1<<"6. ";
for(int i=0;i<20;i++)
{f1<<name6[i];}
f1<<"----------"<<convert(credit6)<<endl;

f1<<"7. ";
for(int i=0;i<20;i++)
{f1<<name7[i];}
f1<<"----------"<<convert(credit7)<<endl;

f1<<"8. ";
for(int i=0;i<20;i++)
{f1<<name8[i];}
```

```
    f1<<"----------"<<convert(credit8)<<endl;


    f1<<"9.  ";
    for(int i=0;i<20;i++)
    {f1<<name9[i];}
    f1<<"----------"<<convert(credit9)<<endl;


    f1<<"10.";
    for(int i=0;i<20;i++)
    {f1<<name[i];}
    f1<<"----------"<<credit<<endl;


    transfer(rcredit1,convert(credit1));
transfer(rcredit2,convert(credit2));
transfer(rcredit3,convert(credit3));
transfer(rcredit4,convert(credit4));
transfer(rcredit5,convert(credit5));
transfer(rcredit6,convert(credit6));
transfer(rcredit7,convert(credit7));
transfer(rcredit8,convert(credit8));
transfer(rcredit9,convert(credit9));
transfer(rcredit0,credit);


ofstream f2("Rank List.txt");
    for(int i=0;i<20;i++)
    {f2<<name1[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit1[i];}
```

```
	for(int i=0;i<20;i++)
	{f2<<name2[i];}
	for(int i=0;i<6;i++)
	{f2<<rcredit2[i];}

	for(int i=0;i<20;i++)
	{f2<<name3[i];}
	for(int i=0;i<6;i++)
	{f2<<rcredit3[i];}

	for(int i=0;i<20;i++)
	{f2<<name4[i];}
	for(int i=0;i<6;i++)
	{f2<<rcredit4[i];}

	for(int i=0;i<20;i++)
	{f2<<name5[i];}
	for(int i=0;i<6;i++)
	{f2<<rcredit5[i];}

	for(int i=0;i<20;i++)
	{f2<<name6[i];}
	for(int i=0;i<6;i++)
	{f2<<rcredit6[i];}

	for(int i=0;i<20;i++)
	{f2<<name7[i];}
	for(int i=0;i<6;i++)
	{f2<<rcredit7[i];}
```

```
for(int i=0;i<20;i++)
{f2<<name8[i];}
for(int i=0;i<6;i++)
{f2<<rcredit8[i];}

for(int i=0;i<20;i++)
{f2<<name9[i];}
for(int i=0;i<6;i++)
{f2<<rcredit9[i];}

for(int i=0;i<20;i++)
{f2<<name[i];}
for(int i=0;i<6;i++)
{f2<<rcredit0[i];}

f2.close();

}
else
{
    if((credit<=convert(credit8))&&(credit>convert(credit9)))
    {
f1<<"1. ";
for(int i=0;i<20;i++)
{f1<<name1[i];}
f1<<"----------"<<convert(credit1)<<endl;

f1<<"2. ";
```

```
for(int i=0;i<20;i++)
{f1<<name2[i];}
f1<<"----------"<<convert(credit2)<<endl;

f1<<"3. ";
for(int i=0;i<20;i++)
{f1<<name3[i];}
f1<<"----------"<<convert(credit3)<<endl;

f1<<"4. ";
for(int i=0;i<20;i++)
{f1<<name4[i];}
f1<<"----------"<<convert(credit4)<<endl;

f1<<"5. ";
for(int i=0;i<20;i++)
{f1<<name5[i];}
f1<<"----------"<<convert(credit5)<<endl;

f1<<"6. ";
for(int i=0;i<20;i++)
{f1<<name6[i];}
f1<<"----------"<<convert(credit6)<<endl;

f1<<"7. ";
for(int i=0;i<20;i++)
{f1<<name7[i];}
f1<<"----------"<<convert(credit7)<<endl;
```

```
    f1<<"8. ";

    for(int i=0;i<20;i++)

    {f1<<name8[i];}

    f1<<"----------"<<convert(credit8)<<endl;


    f1<<"9. ";

    for(int i=0;i<20;i++)

    {f1<<name[i];}

    f1<<"----------"<<credit<<endl;


    f1<<"10.";

    for(int i=0;i<20;i++)

    {f1<<name9[i];}

    f1<<"----------"<<convert(credit9)<<endl;


transfer(rcredit1,convert(credit1));

transfer(rcredit2,convert(credit2));

transfer(rcredit3,convert(credit3));

transfer(rcredit4,convert(credit4));

transfer(rcredit5,convert(credit5));

transfer(rcredit6,convert(credit6));

transfer(rcredit7,convert(credit7));

transfer(rcredit8,convert(credit8));

transfer(rcredit9,credit);

transfer(rcredit0,convert(credit9));


ofstream f2("Rank List.txt");

    for(int i=0;i<20;i++)

    {f2<<name1[i];}
```

```
for(int i=0;i<6;i++)
{f2<<rcredit1[i];}

for(int i=0;i<20;i++)
{f2<<name2[i];}
for(int i=0;i<6;i++)
{f2<<rcredit2[i];}

for(int i=0;i<20;i++)
{f2<<name3[i];}
for(int i=0;i<6;i++)
{f2<<rcredit3[i];}

for(int i=0;i<20;i++)
{f2<<name4[i];}
for(int i=0;i<6;i++)
{f2<<rcredit4[i];}

for(int i=0;i<20;i++)
{f2<<name5[i];}
for(int i=0;i<6;i++)
{f2<<rcredit5[i];}

for(int i=0;i<20;i++)
{f2<<name6[i];}
for(int i=0;i<6;i++)
{f2<<rcredit6[i];}

for(int i=0;i<20;i++)
```

```
{f2<<name7[i];}
for(int i=0;i<6;i++)
{f2<<rcredit7[i];}

for(int i=0;i<20;i++)
{f2<<name8[i];}
for(int i=0;i<6;i++)
{f2<<rcredit8[i];}

for(int i=0;i<20;i++)
{f2<<name[i];}
for(int i=0;i<6;i++)
{f2<<rcredit9[i];}

for(int i=0;i<20;i++)
{f2<<name9[i];}
for(int i=0;i<6;i++)
{f2<<rcredit0[i];}

f2.close();

    }
    else
    {
        if((credit<=convert(credit7))&&(credit>convert(credit8)))
        {
                                f1<<"1. ";
for(int i=0;i<20;i++)
{f1<<name1[i];}
```

```
f1<<"----------"<<convert(credit1)<<endl;


f1<<"2. ";
for(int i=0;i<20;i++)
{f1<<name2[i];}
f1<<"----------"<<convert(credit2)<<endl;


f1<<"3. ";
for(int i=0;i<20;i++)
{f1<<name3[i];}
f1<<"----------"<<convert(credit3)<<endl;


f1<<"4. ";
for(int i=0;i<20;i++)
{f1<<name4[i];}
f1<<"----------"<<convert(credit4)<<endl;


f1<<"5. ";
for(int i=0;i<20;i++)
{f1<<name5[i];}
f1<<"----------"<<convert(credit5)<<endl;


f1<<"6. ";
for(int i=0;i<20;i++)
{f1<<name6[i];}
f1<<"----------"<<convert(credit6)<<endl;


f1<<"7. ";
for(int i=0;i<20;i++)
```

```
    {f1<<name7[i];}
    f1<<"----------"<<convert(credit7)<<endl;


    f1<<"8. ";
    for(int i=0;i<20;i++)
    {f1<<name[i];}
    f1<<"----------"<<credit<<endl;


    f1<<"9. ";
    for(int i=0;i<20;i++)
    {f1<<name8[i];}
    f1<<"----------"<<convert(credit8)<<endl;


    f1<<"10.";
    for(int i=0;i<20;i++)
    {f1<<name9[i];}
    f1<<"----------"<<convert(credit9)<<endl;


transfer(rcredit1,convert(credit1));
transfer(rcredit2,convert(credit2));
transfer(rcredit3,convert(credit3));
transfer(rcredit4,convert(credit4));
transfer(rcredit5,convert(credit5));
transfer(rcredit6,convert(credit6));
transfer(rcredit7,convert(credit7));
transfer(rcredit8,credit);
transfer(rcredit9,convert(credit8));
transfer(rcredit0,convert(credit9));
```

```
ofstream f2("Rank List.txt");
    for(int i=0;i<20;i++)
    {f2<<name1[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit1[i];}

    for(int i=0;i<20;i++)
    {f2<<name2[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit2[i];}

    for(int i=0;i<20;i++)
    {f2<<name3[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit3[i];}

    for(int i=0;i<20;i++)
    {f2<<name4[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit4[i];}

    for(int i=0;i<20;i++)
    {f2<<name5[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit5[i];}

    for(int i=0;i<20;i++)
    {f2<<name6[i];}
    for(int i=0;i<6;i++)
```

```
{f2<<rcredit6[i];}


for(int i=0;i<20;i++)
{f2<<name7[i];}
for(int i=0;i<6;i++)
{f2<<rcredit7[i];}


for(int i=0;i<20;i++)
{f2<<name[i];}
for(int i=0;i<6;i++)
{f2<<rcredit8[i];}


for(int i=0;i<20;i++)
{f2<<name8[i];}
for(int i=0;i<6;i++)
{f2<<rcredit9[i];}


for(int i=0;i<20;i++)
{f2<<name9[i];}
for(int i=0;i<6;i++)
{f2<<rcredit0[i];}


f2.close();



         }
         else
         {
```

```
if((credit<=convert(credit6))&&(credit>convert(credit7)))
              {
f1<<"1.  ";
for(int i=0;i<20;i++)
{f1<<name1[i];}
f1<<"----------"<<convert(credit1)<<endl;

f1<<"2.  ";
for(int i=0;i<20;i++)
{f1<<name2[i];}
f1<<"----------"<<convert(credit2)<<endl;

f1<<"3.  ";
for(int i=0;i<20;i++)
{f1<<name3[i];}
f1<<"----------"<<convert(credit3)<<endl;

f1<<"4.  ";
for(int i=0;i<20;i++)
{f1<<name4[i];}
f1<<"----------"<<convert(credit4)<<endl;

f1<<"5.  ";
for(int i=0;i<20;i++)
{f1<<name5[i];}
f1<<"----------"<<convert(credit5)<<endl;

f1<<"6.  ";
for(int i=0;i<20;i++)
```

```
    {f1<<name6[i];}

    f1<<"----------"<<convert(credit6)<<endl;


    f1<<"7.  ";

    for(int i=0;i<20;i++)

    {f1<<name[i];}

    f1<<"----------"<<credit<<endl;


    f1<<"8.  ";

    for(int i=0;i<20;i++)

    {f1<<name7[i];}

    f1<<"----------"<<convert(credit7)<<endl;


    f1<<"9.  ";

    for(int i=0;i<20;i++)

    {f1<<name8[i];}

    f1<<"----------"<<convert(credit8)<<endl;


    f1<<"10.";

    for(int i=0;i<20;i++)

    {f1<<name9[i];}

    f1<<"----------"<<convert(credit9)<<endl;


transfer(rcredit1,convert(credit1));

transfer(rcredit2,convert(credit2));

transfer(rcredit3,convert(credit3));

transfer(rcredit4,convert(credit4));

transfer(rcredit5,convert(credit5));

transfer(rcredit6,convert(credit6));
transfer(rcredit7,credit);

transfer(rcredit8,convert(credit7));

transfer(rcredit9,convert(credit8));

transfer(rcredit0,convert(credit9));


ofstream f2("Rank List.txt");

    for(int i=0;i<20;i++)

    {f2<<name1[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit1[i];}


    for(int i=0;i<20;i++)

    {f2<<name2[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit2[i];}


    for(int i=0;i<20;i++)

    {f2<<name3[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit3[i];}


    for(int i=0;i<20;i++)

    {f2<<name4[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit4[i];}


    for(int i=0;i<20;i++)

    {f2<<name5[i];}

    for(int i=0;i<6;i++)
    {f2<<rcredit5[i];}


    for(int i=0;i<20;i++)

    {f2<<name6[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit6[i];}


    for(int i=0;i<20;i++)

    {f2<<name[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit7[i];}


    for(int i=0;i<20;i++)

    {f2<<name7[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit8[i];}


    for(int i=0;i<20;i++)

    {f2<<name8[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit9[i];}


    for(int i=0;i<20;i++)

    {f2<<name9[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit0[i];}


    f2.close();

                  }
```

```
                else
                {

   if((credit<=convert(credit5))&&(credit>convert(credit6)))
                    {

       f1<<"1. ";
   for(int i=0;i<20;i++)
   {f1<<name1[i];}
   f1<<"----------"<<convert(credit1)<<endl;

   f1<<"2. ";
   for(int i=0;i<20;i++)
   {f1<<name2[i];}
   f1<<"----------"<<convert(credit2)<<endl;

   f1<<"3. ";
   for(int i=0;i<20;i++)
   {f1<<name3[i];}
   f1<<"----------"<<convert(credit3)<<endl;

   f1<<"4. ";
   for(int i=0;i<20;i++)
   {f1<<name4[i];}
   f1<<"----------"<<convert(credit4)<<endl;

   f1<<"5. ";
   for(int i=0;i<20;i++)
   {f1<<name5[i];}
```

```
    f1<<"----------"<<convert(credit5)<<endl;


    f1<<"6. ";
    for(int i=0;i<20;i++)
    {f1<<name[i];}
    f1<<"----------"<<credit<<endl;


    f1<<"7. ";
    for(int i=0;i<20;i++)
    {f1<<name6[i];}
    f1<<"----------"<<convert(credit6)<<endl;


    f1<<"8. ";
    for(int i=0;i<20;i++)
    {f1<<name7[i];}
    f1<<"----------"<<convert(credit7)<<endl;


    f1<<"9. ";
    for(int i=0;i<20;i++)
    {f1<<name8[i];}
    f1<<"----------"<<convert(credit8)<<endl;


    f1<<"10.";
    for(int i=0;i<20;i++)
    {f1<<name9[i];}
    f1<<"----------"<<convert(credit9)<<endl;


    transfer(rcredit1,convert(credit1));
transfer(rcredit2,convert(credit2));
transfer(rcredit3,convert(credit3));

transfer(rcredit4,convert(credit4));

transfer(rcredit5,convert(credit5));

transfer(rcredit6,credit);

transfer(rcredit7,convert(credit6));

transfer(rcredit8,convert(credit7));

transfer(rcredit9,convert(credit8));

transfer(rcredit0,convert(credit9));


ofstream f2("Rank List.txt");

    for(int i=0;i<20;i++)

    {f2<<name1[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit1[i];}


    for(int i=0;i<20;i++)

    {f2<<name2[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit2[i];}


    for(int i=0;i<20;i++)

    {f2<<name3[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit3[i];}


    for(int i=0;i<20;i++)

    {f2<<name4[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit4[i];}
```

```
for(int i=0;i<20;i++)
{f2<<name5[i];}
for(int i=0;i<6;i++)
{f2<<rcredit5[i];}

for(int i=0;i<20;i++)
{f2<<name[i];}
for(int i=0;i<6;i++)
{f2<<rcredit6[i];}

for(int i=0;i<20;i++)
{f2<<name6[i];}
for(int i=0;i<6;i++)
{f2<<rcredit7[i];}

for(int i=0;i<20;i++)
{f2<<name7[i];}
for(int i=0;i<6;i++)
{f2<<rcredit8[i];}

for(int i=0;i<20;i++)
{f2<<name8[i];}
for(int i=0;i<6;i++)
{f2<<rcredit9[i];}

for(int i=0;i<20;i++)
{f2<<name9[i];}
for(int i=0;i<6;i++)
```

```
{f2<<rcredit0[i];}

f2.close();

                  }
                  else
                  {

if((credit<=convert(credit4))&&(credit>convert(credit5)))
                       {
f1<<"1.  ";
for(int i=0;i<20;i++)
{f1<<name1[i];}
f1<<"----------"<<convert(credit1)<<endl;

f1<<"2.  ";
for(int i=0;i<20;i++)
{f1<<name2[i];}
f1<<"----------"<<convert(credit2)<<endl;

f1<<"3.  ";
for(int i=0;i<20;i++)
{f1<<name3[i];}
f1<<"----------"<<convert(credit3)<<endl;

f1<<"4.  ";
for(int i=0;i<20;i++)
{f1<<name4[i];}
f1<<"----------"<<convert(credit4)<<endl;
```

```
f1<<"5. ";

for(int i=0;i<20;i++)

{f1<<name[i];}

f1<<"----------"<<credit<<endl;


f1<<"6. ";

for(int i=0;i<20;i++)

{f1<<name5[i];}

f1<<"----------"<<convert(credit5)<<endl;


f1<<"7. ";

for(int i=0;i<20;i++)

{f1<<name6[i];}

f1<<"----------"<<convert(credit6)<<endl;


f1<<"8. ";

for(int i=0;i<20;i++)

{f1<<name7[i];}

f1<<"----------"<<convert(credit7)<<endl;


f1<<"9. ";

for(int i=0;i<20;i++)

{f1<<name8[i];}

f1<<"----------"<<convert(credit8)<<endl;


f1<<"10.";

for(int i=0;i<20;i++)

{f1<<name9[i];}
```

```
    f1<<"----------"<<convert(credit9)<<endl;


transfer(rcredit1,convert(credit1));

transfer(rcredit2,convert(credit2));

transfer(rcredit3,convert(credit3));

transfer(rcredit4,convert(credit4));

transfer(rcredit5,credit);

transfer(rcredit6,convert(credit5));

transfer(rcredit7,convert(credit6));

transfer(rcredit8,convert(credit7));

transfer(rcredit9,convert(credit8));

transfer(rcredit0,convert(credit9));


ofstream f2("Rank List.txt");

    for(int i=0;i<20;i++)

    {f2<<name1[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit1[i];}


    for(int i=0;i<20;i++)

    {f2<<name2[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit2[i];}


    for(int i=0;i<20;i++)

    {f2<<name3[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit3[i];}
    for(int i=0;i<20;i++)
    {f2<<name4[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit4[i];}

    for(int i=0;i<20;i++)
    {f2<<name[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit5[i];}

    for(int i=0;i<20;i++)
    {f2<<name5[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit6[i];}

    for(int i=0;i<20;i++)
    {f2<<name6[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit7[i];}

    for(int i=0;i<20;i++)
    {f2<<name7[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit8[i];}

    for(int i=0;i<20;i++)
    {f2<<name8[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit9[i];}
```

```
for(int i=0;i<20;i++)
{f2<<name9[i];}
for(int i=0;i<6;i++)
{f2<<rcredit0[i];}

f2.close();
                      }
                      else
                      {

if((credit<=convert(credit3))&&(credit>convert(credit4)))
                           {
                      f1<<"1. ";
for(int i=0;i<20;i++)
{f1<<name1[i];}
f1<<"----------"<<convert(credit1)<<endl;

f1<<"2. ";
for(int i=0;i<20;i++)
{f1<<name2[i];}
f1<<"----------"<<convert(credit2)<<endl;

f1<<"3. ";
for(int i=0;i<20;i++)
{f1<<name3[i];}
f1<<"----------"<<convert(credit3)<<endl;

f1<<"4. ";
```

```
for(int i=0;i<20;i++)
{f1<<name[i];}
f1<<"----------"<<credit<<endl;

f1<<"5. ";
for(int i=0;i<20;i++)
{f1<<name4[i];}
f1<<"----------"<<convert(credit4)<<endl;

f1<<"6. ";
for(int i=0;i<20;i++)
{f1<<name5[i];}
f1<<"----------"<<convert(credit5)<<endl;

f1<<"7. ";
for(int i=0;i<20;i++)
{f1<<name6[i];}
f1<<"----------"<<convert(credit6)<<endl;

f1<<"8. ";
for(int i=0;i<20;i++)
{f1<<name7[i];}
f1<<"----------"<<convert(credit7)<<endl;

f1<<"9. ";
for(int i=0;i<20;i++)
{f1<<name8[i];}
f1<<"----------"<<convert(credit8)<<endl;
```

```
    f1<<"10.";
    for(int i=0;i<20;i++)
    {f1<<name9[i];}
    f1<<"----------"<<convert(credit9)<<endl;



transfer(rcredit1,convert(credit1));
transfer(rcredit2,convert(credit2));
transfer(rcredit3,convert(credit3));
transfer(rcredit4,credit);
transfer(rcredit5,convert(credit4));
transfer(rcredit6,convert(credit5));
transfer(rcredit7,convert(credit6));
transfer(rcredit8,convert(credit7));
transfer(rcredit9,convert(credit8));
transfer(rcredit0,convert(credit9));

ofstream f2("Rank List.txt");
    for(int i=0;i<20;i++)
    {f2<<name1[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit1[i];}

    for(int i=0;i<20;i++)
    {f2<<name2[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit2[i];}

    for(int i=0;i<20;i++)
```

```
{f2<<name3[i];}
for(int i=0;i<6;i++)
{f2<<rcredit3[i];}

for(int i=0;i<20;i++)
{f2<<name[i];}
for(int i=0;i<6;i++)
{f2<<rcredit4[i];}

for(int i=0;i<20;i++)
{f2<<name4[i];}
for(int i=0;i<6;i++)
{f2<<rcredit5[i];}

for(int i=0;i<20;i++)
{f2<<name5[i];}
for(int i=0;i<6;i++)
{f2<<rcredit6[i];}

for(int i=0;i<20;i++)
{f2<<name6[i];}
for(int i=0;i<6;i++)
{f2<<rcredit7[i];}

for(int i=0;i<20;i++)
{f2<<name7[i];}
for(int i=0;i<6;i++)
{f2<<rcredit8[i];}
```

```
for(int i=0;i<20;i++)

{f2<<name8[i];}

for(int i=0;i<6;i++)

{f2<<rcredit9[i];}



for(int i=0;i<20;i++)

{f2<<name9[i];}

for(int i=0;i<6;i++)

{f2<<rcredit0[i];}



f2.close();



                            }

                            else

                            {



if((credit<=convert(credit2))&&(credit>convert(credit3)))

                                 {

f1<<"1.  ";

for(int i=0;i<20;i++)

{f1<<name1[i];}

f1<<"----------"<<convert(credit1)<<endl;



f1<<"2.  ";

for(int i=0;i<20;i++)

{f1<<name2[i];}

f1<<"----------"<<convert(credit2)<<endl;



f1<<"3.  ";
```

```
for(int i=0;i<20;i++)

{f1<<name[i];}

f1<<"----------"<<credit<<endl;


f1<<"4.  ";

for(int i=0;i<20;i++)

{f1<<name3[i];}

f1<<"----------"<<convert(credit3)<<endl;


f1<<"5.  ";

for(int i=0;i<20;i++)

{f1<<name4[i];}

f1<<"----------"<<convert(credit4)<<endl;


f1<<"6.  ";

for(int i=0;i<20;i++)

{f1<<name5[i];}

f1<<"----------"<<convert(credit5)<<endl;


f1<<"7.  ";

for(int i=0;i<20;i++)

{f1<<name6[i];}

f1<<"----------"<<convert(credit6)<<endl;


f1<<"8.  ";

for(int i=0;i<20;i++)

{f1<<name7[i];}

f1<<"----------"<<convert(credit7)<<endl;
```

```
    f1<<"9.  ";
    for(int i=0;i<20;i++)
    {f1<<name8[i];}
    f1<<"----------"<<convert(credit8)<<endl;

    f1<<"10.";
    for(int i=0;i<20;i++)
    {f1<<name9[i];}
    f1<<"----------"<<convert(credit9)<<endl;

transfer(rcredit1,convert(credit1));
transfer(rcredit2,convert(credit2));
transfer(rcredit3,credit);
transfer(rcredit4,convert(credit3));
transfer(rcredit5,convert(credit4));
transfer(rcredit6,convert(credit5));
transfer(rcredit7,convert(credit6));
transfer(rcredit8,convert(credit7));
transfer(rcredit9,convert(credit8));
transfer(rcredit0,convert(credit9));

ofstream f2("Rank List.txt");
    for(int i=0;i<20;i++)
    {f2<<name1[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit1[i];}

    for(int i=0;i<20;i++)
    {f2<<name2[i];}
```

```
for(int i=0;i<6;i++)

{f2<<rcredit2[i];}


for(int i=0;i<20;i++)

{f2<<name[i];}

for(int i=0;i<6;i++)

{f2<<rcredit3[i];}


for(int i=0;i<20;i++)

{f2<<name3[i];}

for(int i=0;i<6;i++)

{f2<<rcredit4[i];}


for(int i=0;i<20;i++)

{f2<<name4[i];}

for(int i=0;i<6;i++)

{f2<<rcredit5[i];}


for(int i=0;i<20;i++)

{f2<<name5[i];}

for(int i=0;i<6;i++)

{f2<<rcredit6[i];}


for(int i=0;i<20;i++)

{f2<<name6[i];}

for(int i=0;i<6;i++)

{f2<<rcredit7[i];}


for(int i=0;i<20;i++)
```

```
{f2<<name7[i];}
for(int i=0;i<6;i++)
{f2<<rcredit8[i];}

for(int i=0;i<20;i++)
{f2<<name8[i];}
for(int i=0;i<6;i++)
{f2<<rcredit9[i];}

for(int i=0;i<20;i++)
{f2<<name9[i];}
for(int i=0;i<6;i++)
{f2<<rcredit0[i];}

f2.close();



                                }
                                else
                                {


if((credit<=convert(credit1))&&(credit>convert(credit2)))
                                    {
f1<<"1. ";
for(int i=0;i<20;i++)
{f1<<name1[i];}
f1<<"----------"<<convert(credit1)<<endl;

f1<<"2. ";
```

```
for(int i=0;i<20;i++)
{f1<<name[i];}
f1<<"----------"<<credit<<endl;

f1<<"3. ";
for(int i=0;i<20;i++)
{f1<<name2[i];}
f1<<"----------"<<convert(credit2)<<endl;

f1<<"4. ";
for(int i=0;i<20;i++)
{f1<<name3[i];}
f1<<"----------"<<convert(credit3)<<endl;

f1<<"5. ";
for(int i=0;i<20;i++)
{f1<<name4[i];}
f1<<"----------"<<convert(credit4)<<endl;

f1<<"6. ";
for(int i=0;i<20;i++)
{f1<<name5[i];}
f1<<"----------"<<convert(credit5)<<endl;

f1<<"7. ";
for(int i=0;i<20;i++)
{f1<<name6[i];}
f1<<"----------"<<convert(credit6)<<endl;
```

```
    f1<<"8. ";

    for(int i=0;i<20;i++)

    {f1<<name7[i];}

    f1<<"----------"<<convert(credit7)<<endl;


    f1<<"9. ";

    for(int i=0;i<20;i++)

    {f1<<name8[i];}

    f1<<"----------"<<convert(credit8)<<endl;


    f1<<"10.";

    for(int i=0;i<20;i++)

    {f1<<name9[i];}

    f1<<"----------"<<convert(credit9)<<endl;


    transfer(rcredit1,convert(credit1));

transfer(rcredit2,credit);

transfer(rcredit3,convert(credit2));

transfer(rcredit4,convert(credit3));

transfer(rcredit5,convert(credit4));

transfer(rcredit6,convert(credit5));

transfer(rcredit7,convert(credit6));

transfer(rcredit8,convert(credit7));

transfer(rcredit9,convert(credit8));

transfer(rcredit0,convert(credit9));


ofstream f2("Rank List.txt");

    for(int i=0;i<20;i++)

    {f2<<name1[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit1[i];}

    for(int i=0;i<20;i++)
    {f2<<name[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit2[i];}

    for(int i=0;i<20;i++)
    {f2<<name2[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit3[i];}

    for(int i=0;i<20;i++)
    {f2<<name3[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit4[i];}

    for(int i=0;i<20;i++)
    {f2<<name4[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit5[i];}

    for(int i=0;i<20;i++)
    {f2<<name5[i];}
    for(int i=0;i<6;i++)
    {f2<<rcredit6[i];}

    for(int i=0;i<20;i++)
```

```
{f2<<name6[i];}
for(int i=0;i<6;i++)
{f2<<rcredit7[i];}

for(int i=0;i<20;i++)
{f2<<name7[i];}
for(int i=0;i<6;i++)
{f2<<rcredit8[i];}

for(int i=0;i<20;i++)
{f2<<name8[i];}
for(int i=0;i<6;i++)
{f2<<rcredit9[i];}

for(int i=0;i<20;i++)
{f2<<name9[i];}
for(int i=0;i<6;i++)
{f2<<rcredit0[i];}

f2.close();

                                    }
                                    else
                                    {
                                        if(credit>convert(credit1))
                                        {
f1<<"1. ";
for(int i=0;i<20;i++)
{f1<<name[i];}
```

```
f1<<"----------"<<credit<<endl;


f1<<"2. ";

for(int i=0;i<20;i++)

{f1<<name1[i];}

f1<<"----------"<<convert(credit1)<<endl;


f1<<"3. ";

for(int i=0;i<20;i++)

{f1<<name2[i];}

f1<<"----------"<<convert(credit2)<<endl;


f1<<"4. ";

for(int i=0;i<20;i++)

{f1<<name3[i];}

f1<<"----------"<<convert(credit3)<<endl;


f1<<"5. ";

for(int i=0;i<20;i++)

{f1<<name4[i];}

f1<<"----------"<<convert(credit4)<<endl;


f1<<"6. ";

for(int i=0;i<20;i++)

{f1<<name5[i];}

f1<<"----------"<<convert(credit5)<<endl;


f1<<"7. ";

for(int i=0;i<20;i++)
```

```
    {f1<<name6[i];}
    f1<<"----------"<<convert(credit6)<<endl;


    f1<<"8. ";
    for(int i=0;i<20;i++)
    {f1<<name7[i];}
    f1<<"----------"<<convert(credit7)<<endl;


    f1<<"9. ";
    for(int i=0;i<20;i++)
    {f1<<name8[i];}
    f1<<"----------"<<convert(credit8)<<endl;


    f1<<"10.";
    for(int i=0;i<20;i++)
    {f1<<name9[i];}
    f1<<"----------"<<convert(credit9)<<endl;


    transfer(rcredit1,credit);
transfer(rcredit2,convert(credit1));
transfer(rcredit3,convert(credit2));
transfer(rcredit4,convert(credit3));
transfer(rcredit5,convert(credit4));
transfer(rcredit6,convert(credit5));
transfer(rcredit7,convert(credit6));
transfer(rcredit8,convert(credit7));
transfer(rcredit9,convert(credit8));
transfer(rcredit0,convert(credit9));
```

```
ofstream f2("Rank List.txt");

    for(int i=0;i<20;i++)

    {f2<<name[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit1[i];}


    for(int i=0;i<20;i++)

    {f2<<name1[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit2[i];}


    for(int i=0;i<20;i++)

    {f2<<name2[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit3[i];}


    for(int i=0;i<20;i++)

    {f2<<name3[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit4[i];}


    for(int i=0;i<20;i++)

    {f2<<name4[i];}

    for(int i=0;i<6;i++)

    {f2<<rcredit5[i];}


    for(int i=0;i<20;i++)

    {f2<<name5[i];}

    for(int i=0;i<6;i++)
```

```
{f2<<rcredit6[i];}


for(int i=0;i<20;i++)

{f2<<name6[i];}

for(int i=0;i<6;i++)

{f2<<rcredit7[i];}


for(int i=0;i<20;i++)

{f2<<name7[i];}

for(int i=0;i<6;i++)

{f2<<rcredit8[i];}


for(int i=0;i<20;i++)

{f2<<name8[i];}

for(int i=0;i<6;i++)

{f2<<rcredit9[i];}


for(int i=0;i<20;i++)

{f2<<name9[i];}

for(int i=0;i<6;i++)

{f2<<rcredit0[i];}


f2.close();
                                             }
                                        }
                                   }
                              }
                         }
                    }
```

```
                                        }
                              }
                    }
          }
}


}
```

3. class showcard

Show the address of the specific card using different numbers.

```
cardshow.h
#include<iostream>
using namespace std;
#include<fstream>
#include<string.h>
class showcard
{
public:
          showcard();// constructor
          // initialize each of the character to a specific address.
          char * address(int b);//main function
          //Enter an number, it would return an address.
          //The number is created randomly.
          private:
                    //Each of the character will return an address of the card. Each of the card
would be represented by one number.
```

char *c1; char *c2; char *c3; char *c4;
char *c5; char *c6; char *c7; char *c8;
char *c9; char *c10; char *cj; char *ck;
char *cq;
	club

char *d1; char *d2; char *d3; char *d4;
char *d5; char *d6; char *d7; char *d8;
char *d9; char *d10; char *dj; char *dk;
char *dq;
	diamond

char *s1; char *s2; char *s3; char *s4;
char *s5; char *s6; char *s7; char *s8;
char *s9; char *s10; char *sj; char *sk;
char *sq;
	spade

char *h1; char *h2; char *h3; char *h4;
char *h5; char *h6; char *h7; char *h8;
char *h9; char *h10; char *hj; char *hk;
char *hq;
	heart

char *jr;
};
	joker

Cardshow.cpp
#include"cardshow.h"
#include<fstream>
#include<iostream>
#include<string.h>
using namespace std;	

```
showcard::showcard()

{

s1="cards\\s1.gif"; s2="cards\\s2.gif";

s3="cards\\s3.gif";  s4="cards\\s4.gif";

s5="cards\\s5.gif";  s6="cards\\s6.gif";  s7="cards\\s7.gif";

s8="cards\\s8.gif";

s9="cards\\s9.gif"; s10="cards\\s10.gif";

sj="cards\\sj.gif"; sk="cards\\sk.gif";

sq="cards\\sq.gif";


c1="cards\\c1.gif"; c2="cards\\c2.gif";

c3="cards\\c3.gif"; c4="cards\\c4.gif";

c5="cards\\c5.gif"; c6="cards\\c6.gif";

c7="cards\\c7.gif"; c8="cards\\c8.gif";

c9="cards\\c9.gif"; c10="cards\\c10.gif";

cj="cards\\cj.gif"; ck="cards\\ck.gif";

cq="cards\\cq.gif";


d1="cards\\d1.gif"; d2="cards\\d2.gif";

d3="cards\\d3.gif"; d4="cards\\d4.gif";

d5="cards\\d5.gif"; d6="cards\\d6.gif";

d7="cards\\d7.gif"; d8="cards\\d8.gif";

d9="cards\\d9.gif"; d10="cards\\d10.gif";

dj="cards\\dj.gif"; dk="cards\\dk.gif";

dq="cards\\dq.gif"; h1="cards\\h1.gif";

h2="cards\\h2.gif"; h3="cards\\h3.gif";

h4="cards\\h4.gif"; h5="cards\\h5.gif";

h6="cards\\h6.gif"; h7="cards\\h7.gif";

h8="cards\\h8.gif"; h9="cards\\h9.gif";
```
	Set the address to the card

```
h10="cards\\h10.gif"; hj="cards\\hj.gif";

hk="cards\\hk.gif"; hq="cards\\hq.gif";


jr="cards\\jr.gif";


}

char* showcard::address(int b){

if (b==0)return c2;if (b==1)return c3;

if (b==2)return c4;if (b==3)return c5;

if (b==4)return c6;if (b==5)return c7;

if (b==6)return c8;if (b==7)return c9;

if (b==8)return c10;if (b==9)return cj;

if (b==10)return cq;if (b==11)return ck;

if (b==12)return c1;


if (b==39)return s2;if (b==40)return s3;

if (b==41)return s4;if (b==42)return s5;

if (b==43)return s6;if (b==44)return s7;

if (b==45)return s8;if (b==46)return s9;

if (b==47)return s10;if (b==48)return sj;

if (b==49)return sq;if (b==50)return sk;

if (b==51)return s1;


if (b==52)return jr;


if (b==26)return h2;if (b==27)return h3;

if (b==28)return h4;if (b==29)return h5;

if (b==30)return h6;if (b==31)return h7;

if (b==32)return h8;if (b==33)return h9;
```

c

`if (b==34)return h10;if (b==35)return hj;`	
`if (b==36)return hq;if (b==37)return hk;`	
`if (b==38)return h1;`	
`if (b==13)return d2;if (b==14)return d3;`	
`if (b==15)return d4;if (b==16)return d5;`	
`if (b==17)return d6;if (b==18)return d7;`	
`if (b==19)return d8;if (b==20)return d9;`	
`if (b==21)return d10;if (b==22)return dj;`	
`if (b==23)return dq;if (b==24)return dk;`	
`if (b==25)return d1;`	
`else`	
`return 0;}`	

4. class namepassword1 and 5. class namepassword

Nameandpasswordcompare1

`Nameandpassword.h`	
`#include<iostream>`	
`using namespace std;`	
`#include<fstream>`	
`#include"string"`	
`class namepassword1`	
`{`	
`public: namepassword1(char*a,char*b);`	Store two characters a,b to the charaters one and two.
`        void initial(char*a,char*b);`	Create an new txt file whose naem is the

	username(a), the first 20 characters are the password(b)
bool compare(char*a,char*b);	Compare the password to the txt file.
char* getname();	Get the name from the file.
int getscore();	Get the score from the file.
private: char one[20];	the username
char two[20];	the password
char ch[20];	get the password from the file
};	
Nameandpassword.cpp	
#include"nameandpassword.h"	
namepassword1::namepassword1(char*a,char*b)	
{	
strcpy(one,a);	
strcpy(two,b);	
}	
void namepassword1::initial(char*a,char*b)	Initialize the username and the password,
{strcat(one,".txt");	store the information in one txt file.
ofstream m(one);	
m<<two;	
m.seekp(20);	
char mm[7]={"000100"};	Set the initial score to 100.
m<<mm;	

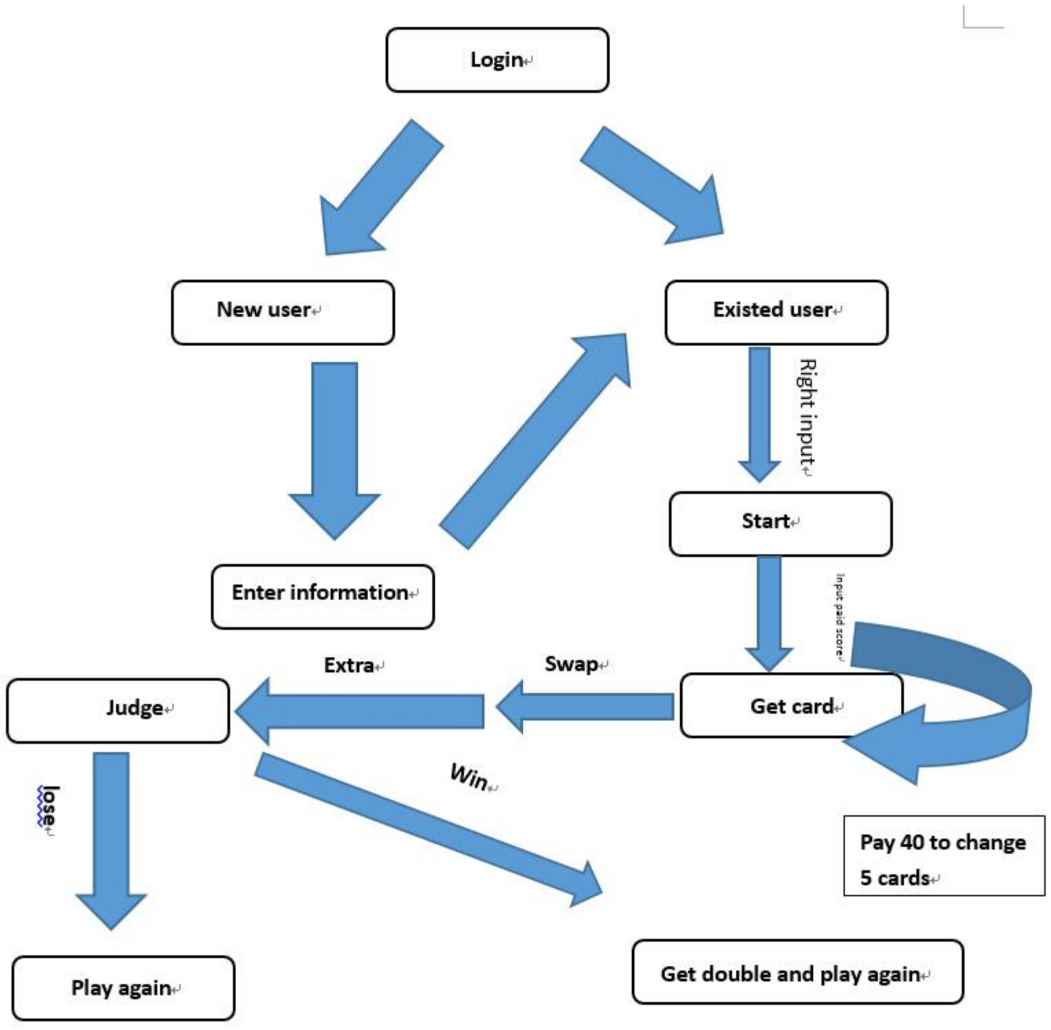
`m.close();`	
`}`	
`char*namepassword1::getname()`	
`{return one;}`	
`int namepassword1::getscore()`	get the score from the file,
`{strcat(one,".txt");`	
`    ifstream q(one);`	
`q.seekg(20);`	
`char n[6];`	
`q>>n;`	
`int m=atoi(n);`	Transfer the character to the integer.
`return m;}`	
`bool namepassword1::compare(char*a,char*b)`	Compare the username and the password.
`{strcat(a,".txt");`	
`    ifstream n(a);`	
`    n.getline(ch,20);`	
`    n.close();`	
`    if (strcmp(ch,b)!=0)`	If there are some differences, return true.
`    {`	
`        return true;}`	
`    else`	If there is no difference, return false.
`    {return false;`	
`    }`	
`}`	
`compare.h`	
`#include<iostream>`	
`using namespace std;`	This library is similar to the library above.
`#include<fstream>`	Since we cannot use the same library in
`#include"string"`	different forms. We created two different

```
class namepassword
{
public: namepassword(char*a,char*b);//store two
characters a,b to the charaters one and two.
        void initial(char*a,char*b);//creat an new txt
file whose naem is the username(a), the first 20
characters are the password.(b)
        bool compare(char*a,char*b);//compare the
password to the txt file.
        char* getname();//get the name from the file.
        int getscore();//get the score from the file.
private: char one[20];//the username
         char two[20];//the password
        char ch[20];//get the password from the file
};
Namepassword.cpp
#include"compare.h"
namepassword::namepassword(char*a,char*b)
{
strcpy(one,a);
strcpy(two,b);
}

void namepassword::initial(char*a,char*b)
{strcat(one,".txt");
    ofstream m(one);
m<<two<<endl;
m.close();
```
	libraries dealing with usernames and passwords.

}	
char*namepassword::getname()	
{return one;}	
int namepassword::getscore()	
{strcat(one,".txt");	
ifstream q(one);	
q.seekg(20);	
char n[6];	Create the txt file storing the username and
q>>n;	the password
int m=atoi(n);	
return m;}	
bool namepassword::compare(char*a,char*b)	
{strcat(a,".txt");	
ifstream n(a);	
n.getline(ch,20);	
n.close();	Get the score from the file.
if (strcmp(ch,b)!=0)	
{	
return true;}	
else	
{return false;	
}	
}	

2.3.2 The Flow of Execution



2.4 Problem Encounter

(1) When changing the character to integer, we found some problems. In GUI, we want to change {“000300”} to integer. The result is 0. We did not know the reason. But after butting “atoi” function into the library, the problem was solved.

(2) At first, we did not know how to swap the cards. After analyzing the conditions, we found out we could use a variable to represent the number of the cards selected. If the user click the card, it would become downward which means it is selected. If the user click the card again, it would become upward and the number of cards selected would be deducted.

(3) When compiling the program function of the rank list, I find that when I output the content of a “.txt” file, the array I save in will appear messy code for the latter digit (The blank digit). So I create another array to strncpy it. When output the content of this array, I just output

the digit what I need.

(4) When compiling the class of judgement, I find that it is difficult to realized judge 6 cards by five cards rules. I try to edit new rules aimed to six cards. However, I come up with a new idea that I can choose each five cards of six cards. Then I can use the rule for five cards to judge. Then I judge six cards by using five times judgement of five cards.

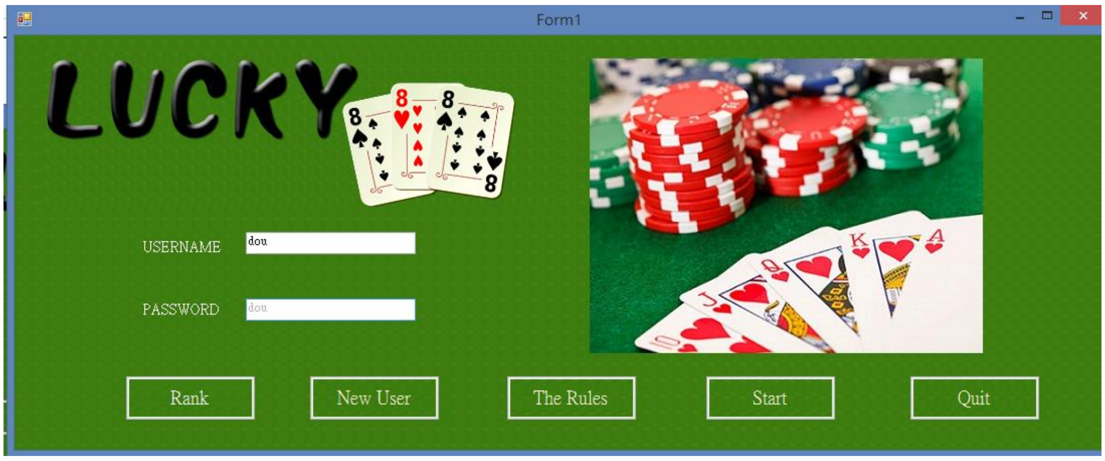
2.5 The way to validate the program

(1) When we first test, we found out the game can only play once. And then, we put the initialization in the button” Start”.

(2) About saving the score and reading the scores from the file, we tested so many times. At last, we put zeros in from of the integers and change the integers to characters to store.

(3) When we tested the rank list, we found out some names could not be read. After looking at the txt file, we found out that there are some unexpected spaces in the characters. So we put some ‘-’ to replace the spaces to get the correct list.

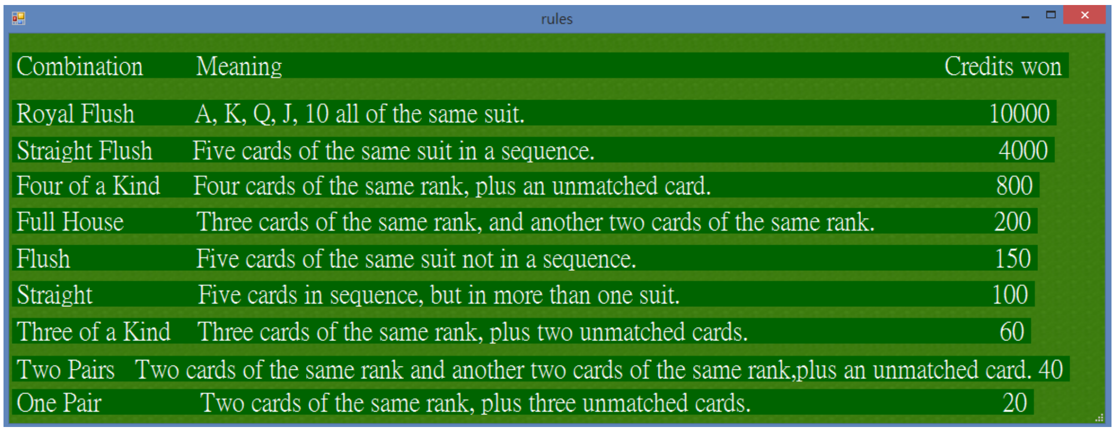
3. <u>Results</u>



USERNAME dou

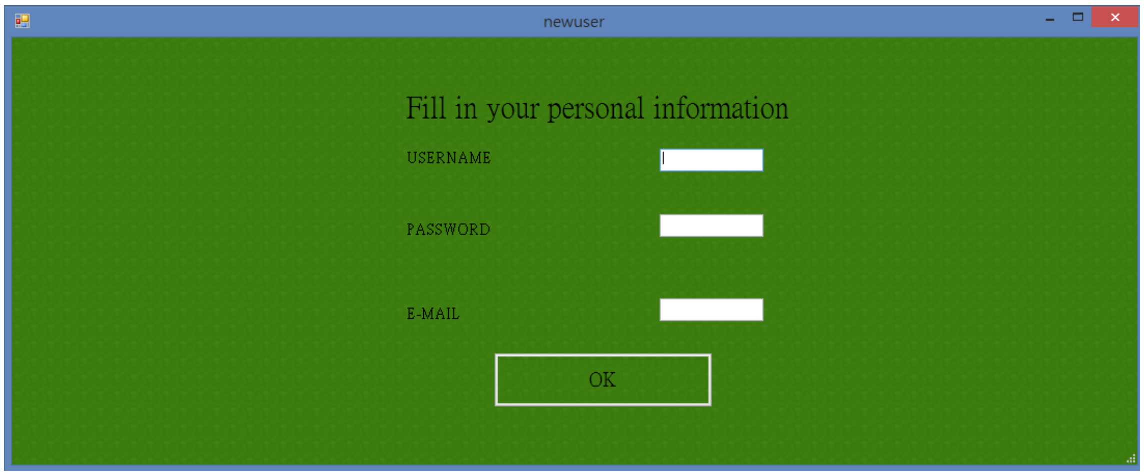
PASSWORD dou

The main menu of our game



Combination	Meaning	Credits won
Royal Flush	A, K, Q, J, 10 all of the same suit.	10000
Straight Flush	Five cards of the same suit in a sequence.	4000
Four of a Kind	Four cards of the same rank, plus an unmatched card.	800
Full House	Three cards of the same rank, and another two cards of the same rank.	200
Flush	Five cards of the same suit not in a sequence.	150
Straight	Five cards in sequence, but in more than one suit.	100
Three of a Kind	Three cards of the same rank, plus two unmatched cards.	60
Two Pairs	Two cards of the same rank and another two cards of the same rank,plus an unmatched card.	40
One Pair	Two cards of the same rank, plus three unmatched cards.	20

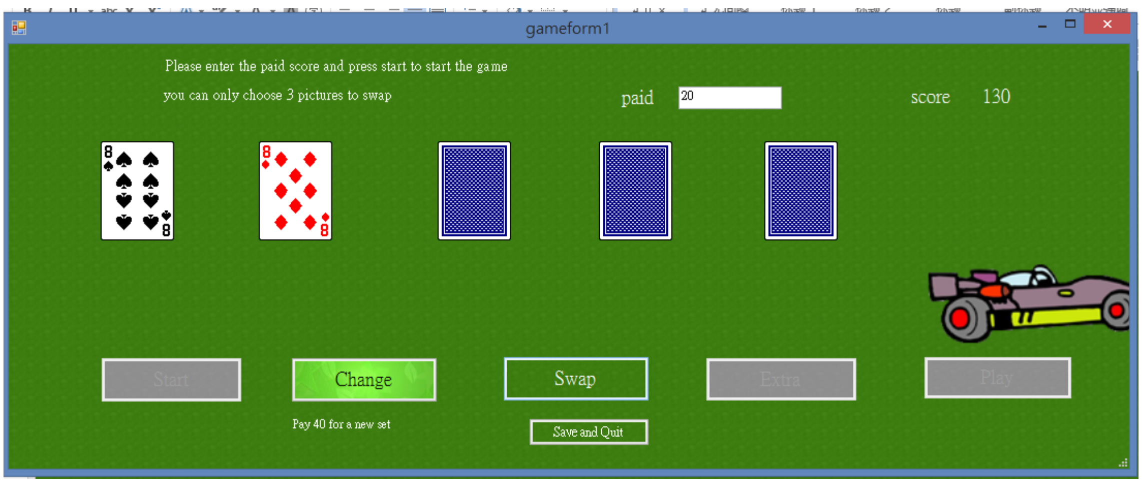
The rules about the game



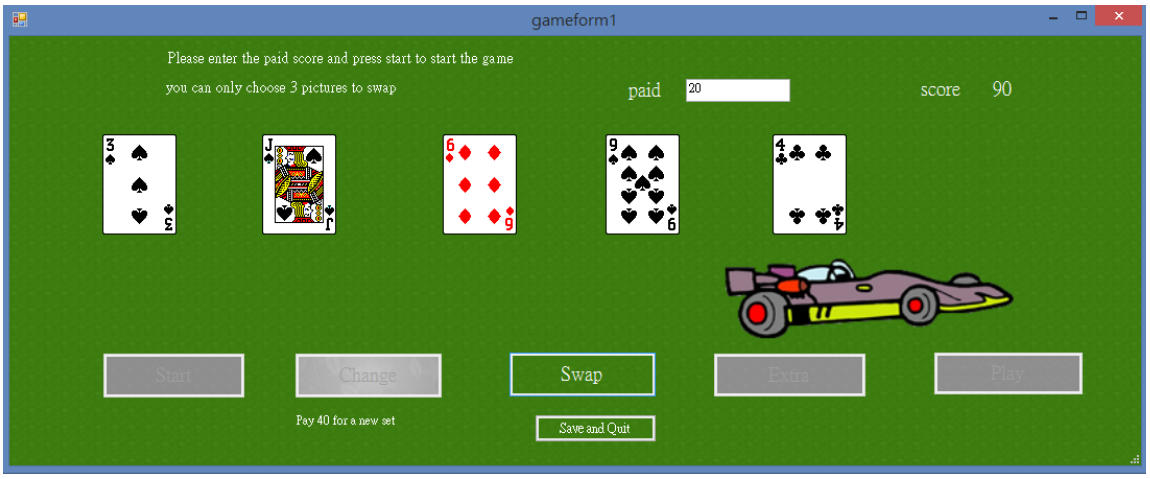
Fill in your personal information

USERNAME	
PASSWORD	
E-MAIL	

The new user form


gameform1
Please enter the paid score and press start to start the game
you can only choose 3 pictures to swap
paid 20
score 130
Start
Change
Swap
Extra
Play
Pay 40 for a new set
Save and Quit

After logging in, the user would enter the score (should be greater or equal to 20), and then, he can swap 0, 1, 2 or 3 cards


gameform1

Please enter the paid score and press start to start the game

you can only choose 3 pictures to swap

paid 20

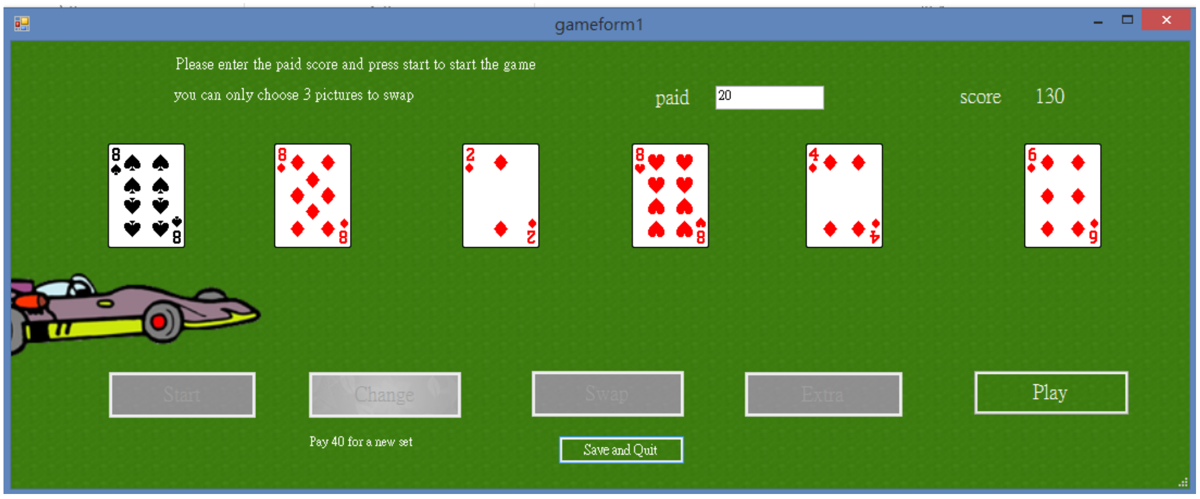
score 90

Start Change Swap Extra Play

Pay 40 for a new set

Save and Quit

If the user do not want this set of cards, he would pay 40 for a new set.


gameform1

Please enter the paid score and press start to start the game

you can only choose 3 pictures to swap

paid 20

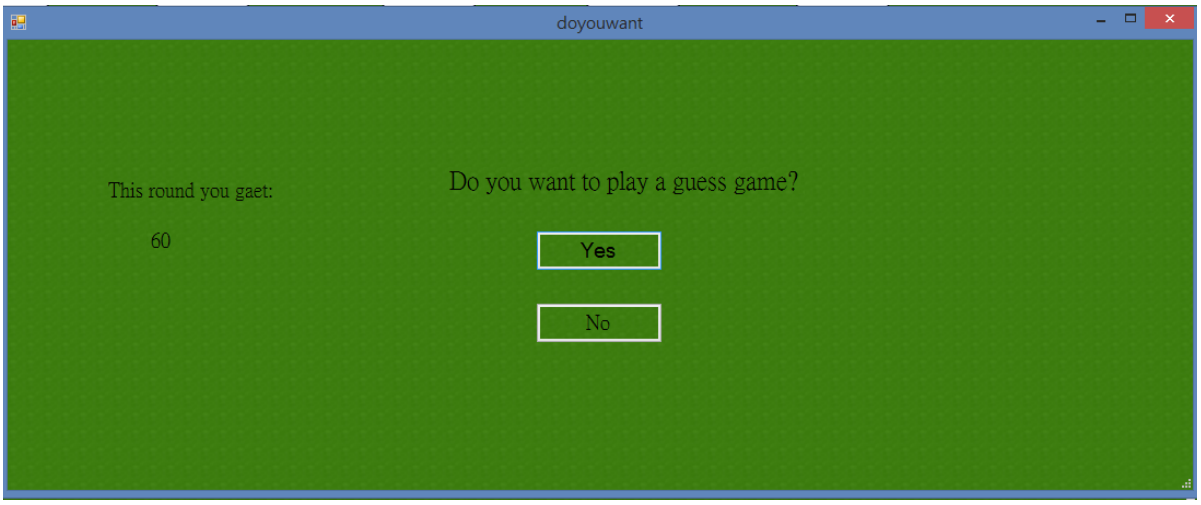
score 130

Start Change Swap Extra Play

Pay 40 for a new set

Save and Quit

After clicking “Swap”, the user could click “Play” to start judging.


doyouwant

This round you gaet:

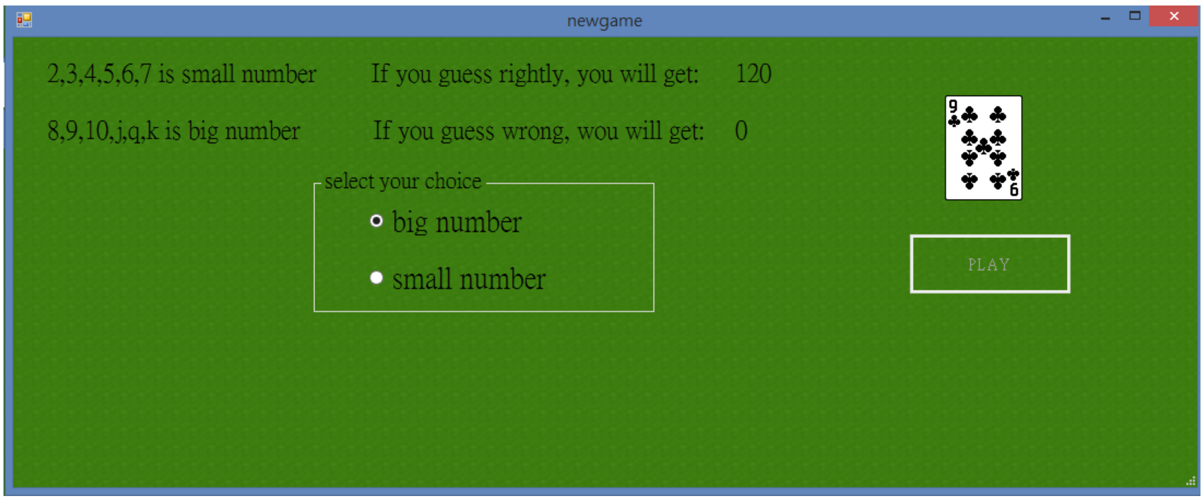
60

Do you want to play a guess game?

Yes

No

If he win, he could choose to play another game or not.


newgame

2,3,4,5,6,7 is small number   If you guess rightly, you will get: 120

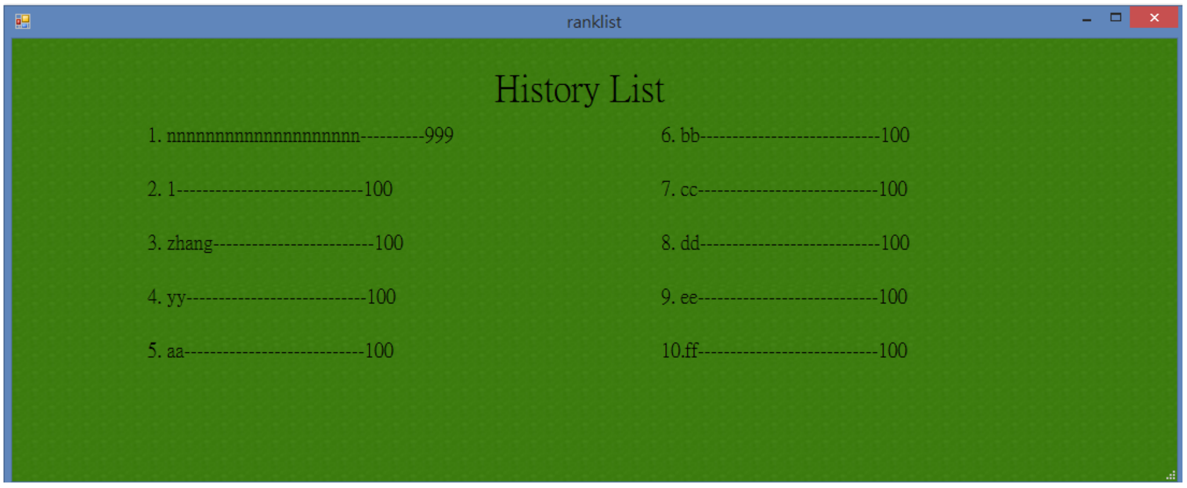
8,9,10,j,q,k is big number   If you guess wrong, wou will get: 0

select your choice

- big number
- small number

PLAY

The user could select the choice. And then, the user click "Play" to play this game.


ranklist

History List

1. nnnnnnnnnnnnnnnnnnnn----------999
2. 1----------------------------100
3. zhang------------------------100
4. yy---------------------------100
5. aa---------------------------100
6. bb---------------------------100
7. cc---------------------------100
8. dd---------------------------100
9. ee---------------------------100
10.ff---------------------------100

The history score list.

4. Conclusion and further development

a) Experience gained in the mini-project

After finishing the programming of mini-project, we gained a lot of about C++ compiling and GUI design. Besides, we think more about the further improvement about our program. When I firstly heard that we will compile a game by ourselves, we are both amazing and exciting. We think that we eventually have the chance to compile our own and entire program by ourselves. During one year studying about C++, we have an initial understanding about the programming. Firstly, we cannot use the class and functions expertly and we have many compiling and running errors. Later on, we gradually know some tips about programming, which cannot be learned from the book. During the programming, we acquire a deeper understanding of C++ compiling such as debugging and managing the codes. What's more, when we encounter the outcomes, we do not give up and search the Internet or ask teachers for help. Besides, we also communicate with our classmates about the compiling skills and the debug we face. In the process of compile the game, we train our brain, have a new aspect about computer programming and abilities to solve problems. We learned that so long as we paid effort and sweat, we will harvest and gain a lot of knowledge about dealing with problems. This programming is one of the most deeply impressive experiment in my life. In conclusion, both of us think that our C++ knowledge has been improved significantly and gain the ability of compile C++ program.

b) Further Improvement

Though we pay a lot effort in our program, because of the limitation of time, there is still a lot of place to improve in the future. We will show what we think it is needed to be improved below.

(1) Add AI to realize PVE in the game. We hope that player can play this game with computer with various difficulties. Computer can have its judge rule, such as calculating the probability of different choice, to gain more credits and win the game. Player can get more fun playing the Lucky888.

(2) Make an improvement for log in system. For example, if the user forget the password, user can reset the password by validating e-mail. If the user log in last time, the system can remember his/her password for the next login.

(3) Increase new game props to increase the interest of the game, such as double-credit props (after using this prop, the credit will be doubled); change card props (after using this prop, the player can change one card randomly). The player need to pay much more credits to buy these props.

(4) This game can add a timer to control the length of time of playing the game. This can avoid the addiction of this game.

Appendix of User Manual

1. Game Rules:

The player can either start a new game or continue a previous game by loading the player's profile. A new game will assign 100 credits to the player initially; otherwise, the number of credits will be loaded from the profile. In each turn, a hand of poker cards will be distributed for every 20 credits paid, where no identical cards (same rank and same suit) will appear in the same hand. The player is allowed to pay more than 20 credits for a new set of cards. The player can also pay 40 credits to change all five cards given. The credits won will then be proportional to the credits paid. The player is allowed to swap cards once. After the player has finished selecting all the cards he wants to swap, a "SWAP" button should be clicked and the new cards (must be different from the cards being replaced) are distributed to replace the selected card(s). After the swapping, the player should click an "EXTRA" button to get extra card(s) for the final hand. The player will win credits if the final hand hits either one of the following combination:

Combination	**Meaning**	**Credits won**
Royal Flush	A, K, Q, J, 10 all of the same suit.	10000
Straight Flush	Five cards of the same suit in a sequence.	4000
Four of a Kind	Four cards of the same rank, plus an unmatched card.	800
Full House	Three cards of the same rank, and another two cards of the same rank.	200
Flush	Five cards of the same suit not in a sequence.	150
Straight	Five cards in sequence, but in more than one suit.	100
Three of a Kind	Three cards of the same rank, plus two unmatched cards.	60
Two Pairs	Two cards of the same rank and another two cards of the same rank, plus an unmatched card.	40
One Pair	Two cards of the same rank, plus three unmatched cards.	20

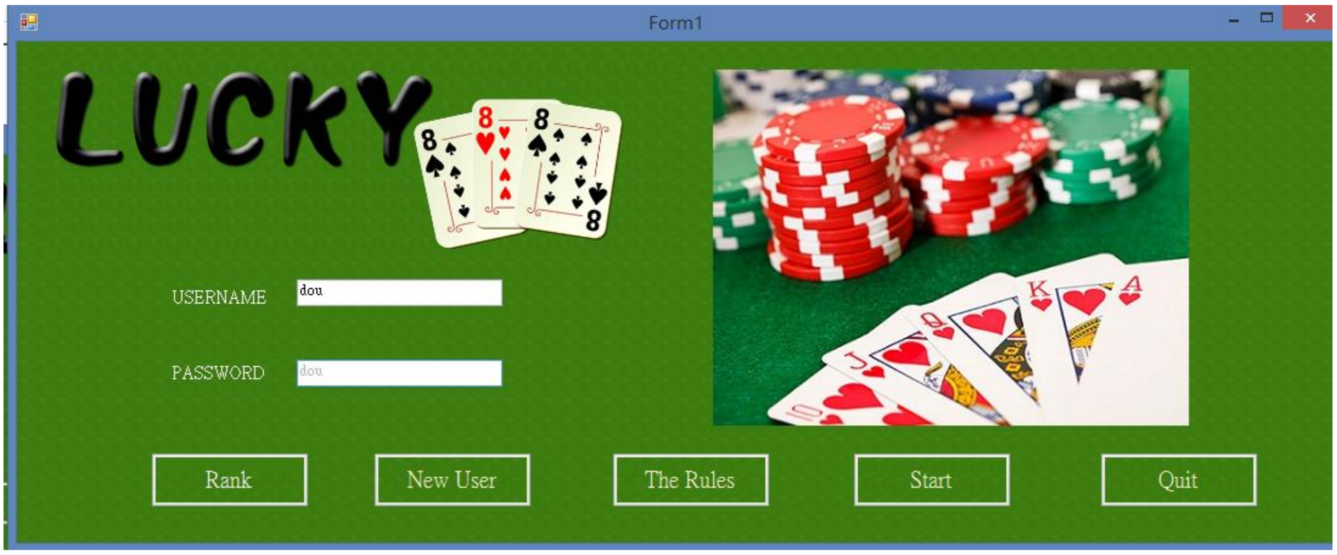
(A joker card is as the wildcard that can represent any card.)

The game ends when the number of credits left is not sufficient to pay for a new set of cards, or the player click a "SAVE" button to quit the game. In the former case, the player's profile will be removed. In the latter case, the credits left will be stored in the player's profile.

In each turn, a hand of five poker cards will be distributed. The player is allowed to swap at most THREE cards for one time. Whenever a card is selected by clicking it, the card turns facedown. The player can unselect the card by clicking the card once more. After the swapping has been done, the player should click the "EXTRA" button to get one more card for the final hand. If the player wins some credits in one turn, he/she is allowed to use the credits won to play a Double-or-None game, say, guessing big or small for an upcoming card. If the player wins this game, the credits would be doubled; otherwise, the credits won would be lost.

2. User Manual of pictures:

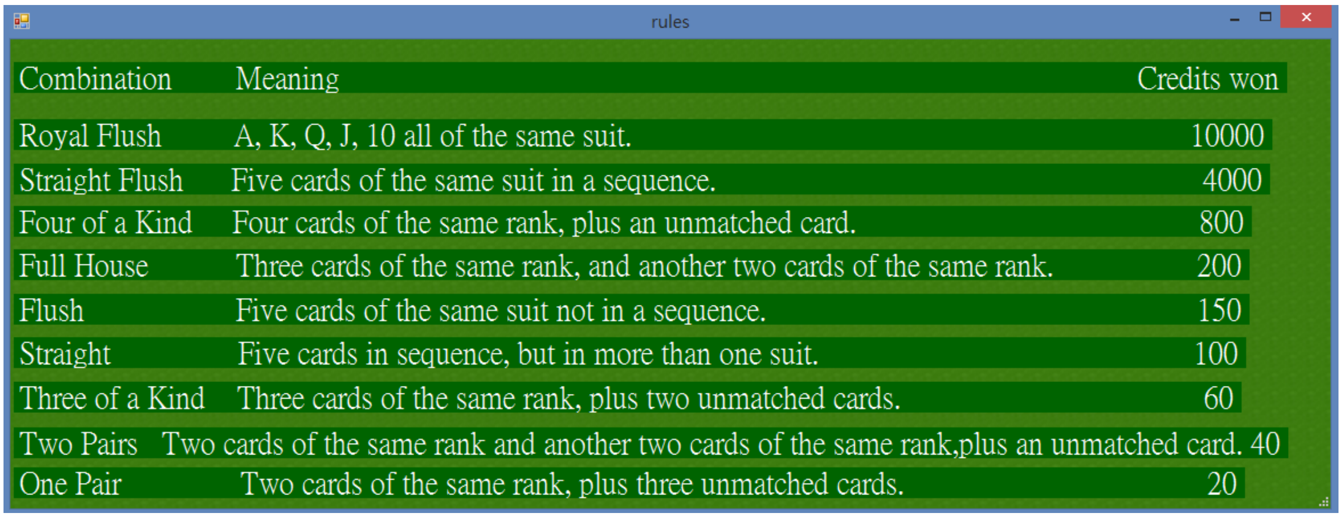
You can log in the game with the original name and password. If you play the game for the first time, please click New User Button.



USERNAME dou

PASSWORD dou

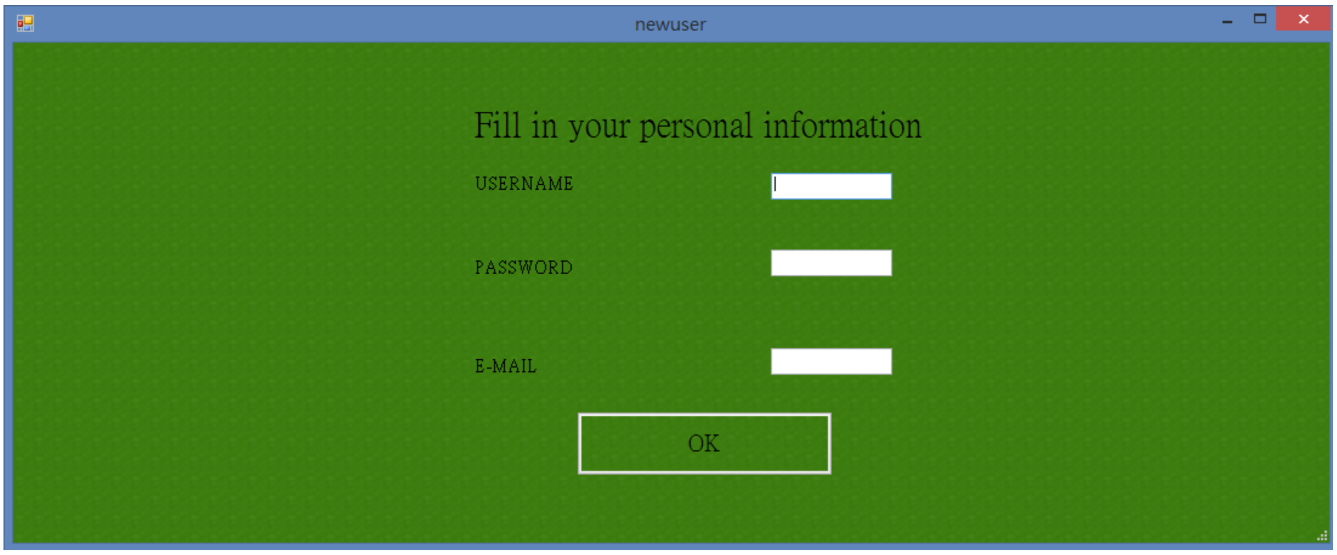
Click the RULE button to acknowledge the rule of Luck 888



rules

Combination	Meaning	Credits won
Royal Flush	A, K, Q, J, 10 all of the same suit.	10000
Straight Flush	Five cards of the same suit in a sequence.	4000
Four of a Kind	Four cards of the same rank, plus an unmatched card.	800
Full House	Three cards of the same rank, and another two cards of the same rank.	200
Flush	Five cards of the same suit not in a sequence.	150
Straight	Five cards in sequence, but in more than one suit.	100
Three of a Kind	Three cards of the same rank, plus two unmatched cards.	60
Two Pairs	Two cards of the same rank and another two cards of the same rank,plus an unmatched card.	40
One Pair	Two cards of the same rank, plus three unmatched cards.	20

If you are a new player for this game, you can create your own profile. (Name for 15 digits and Password for 15 digits)



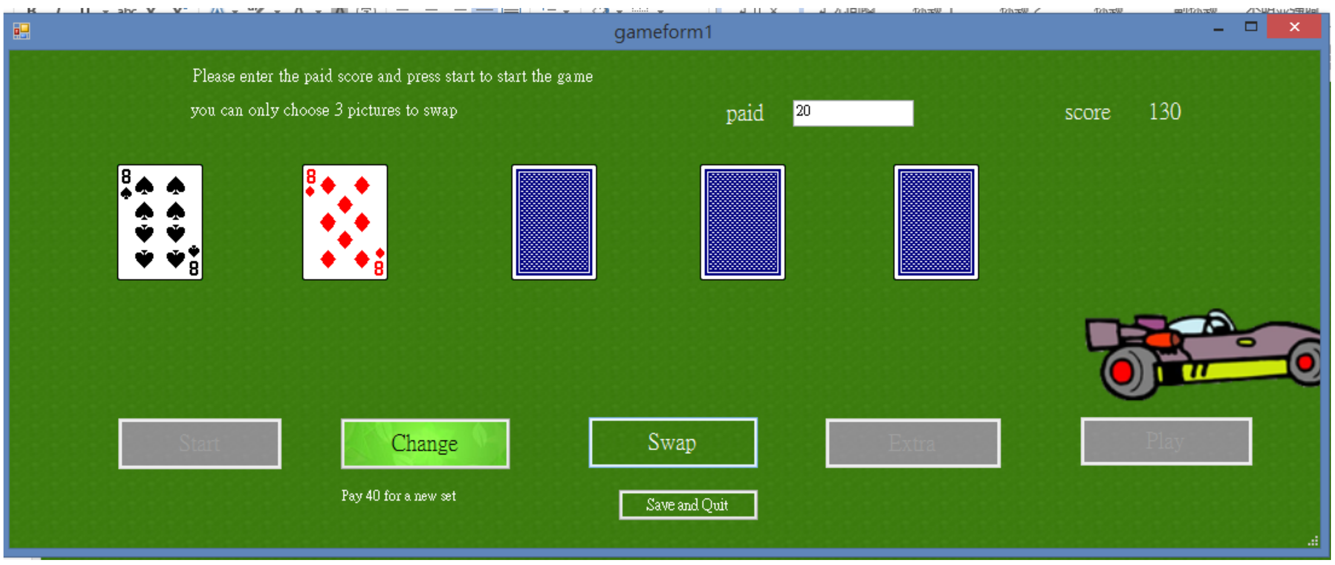
newuser

Fill in your personal information

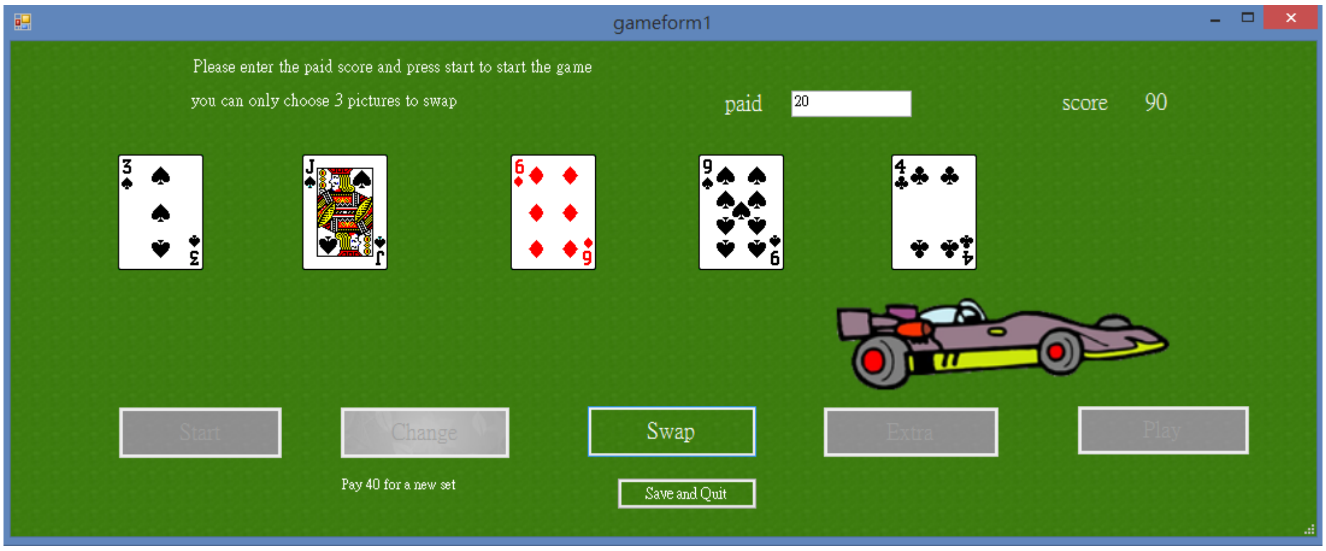
USERNAME	
PASSWORD	
E-MAIL	

You can press Button Change to pay no less than 20 credits to play the game. (The credits won will then be proportional to the credits paid.)

You can pay 40 credits for a new set.


gameform1
Please enter the paid score and press start to start the game
you can only choose 3 pictures to swap
paid
20
score
130
Start
Change
Swap
Extra
Play
Pay 40 for a new set
Save and Quit

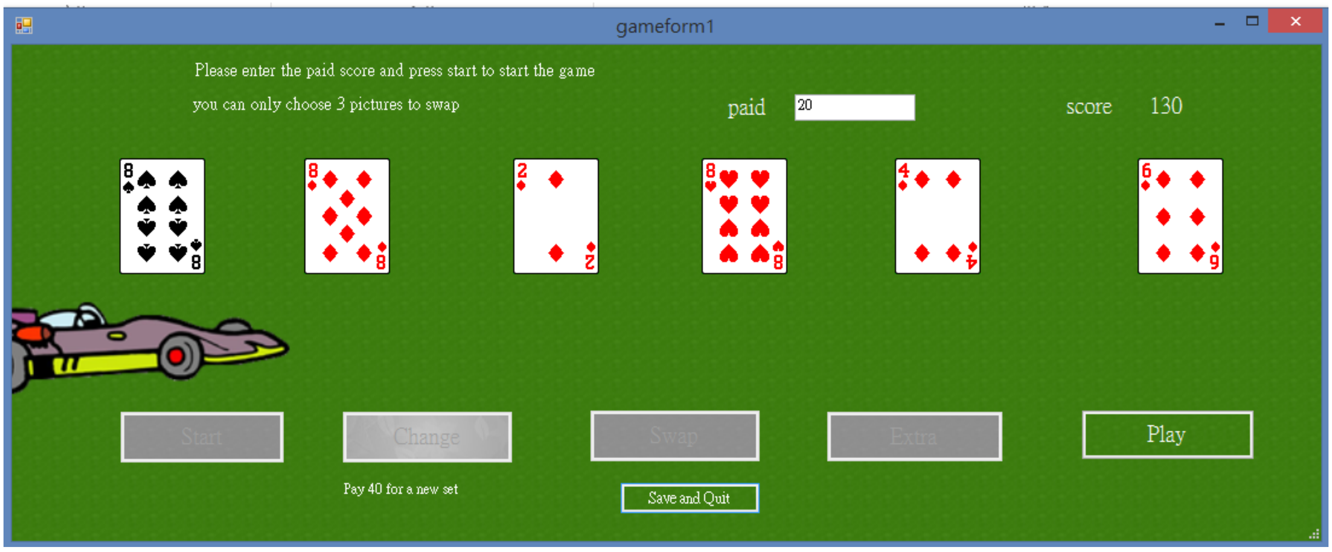
You can press Button Swap to choose any one or two or three cards of five cards to swap.


gameform1
Please enter the paid score and press start to start the game
you can only choose 3 pictures to swap
paid
20
score
90
Start
Change
Swap
Extra
Play
Pay 40 for a new set
Save and Quit

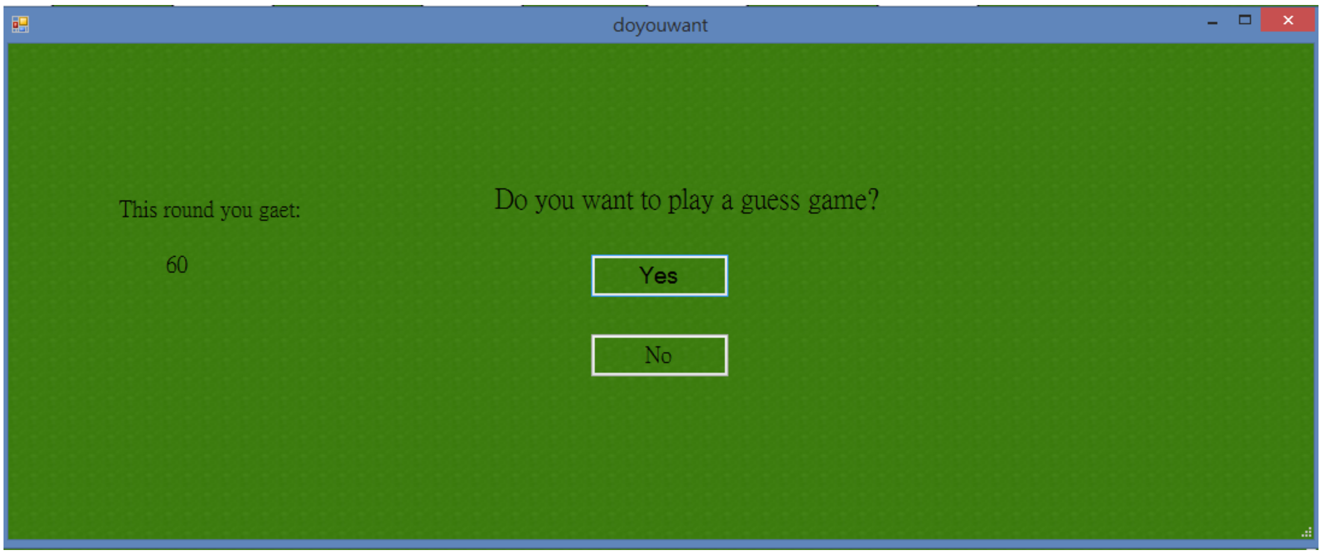
Then you can press Button Extra to get another cards.

You will totally have six cards.

Then you press Button Play to play the game.


gameform1
Please enter the paid score and press start to start the game
you can only choose 3 pictures to swap
paid
20
score
130
Start
Change
Swap
Extra
Play
Pay 40 for a new set
Save and Quit

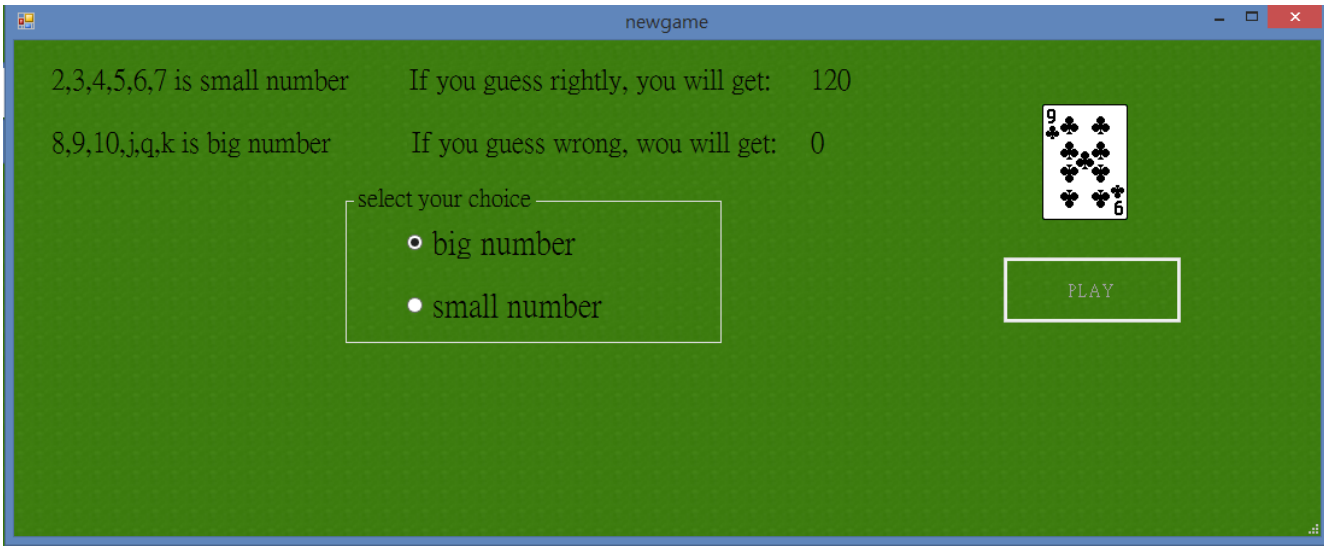
If you win the game, you can play another game to win extra credits. (Guess Game to guess big or small of a random card.)



You can choose big number of small number to play the extra game.

If you win, your credit will be doubled.

If you lose, your earned credit will be zero.



newgame

2,3,4,5,6,7 is small number

8,9,10,j,q,k is big number

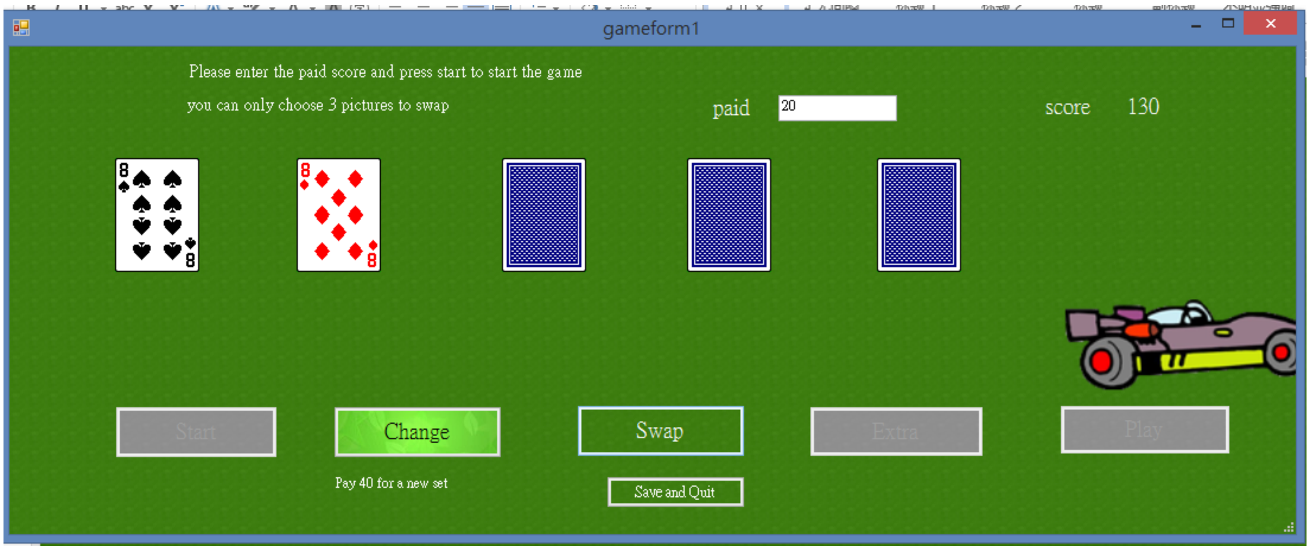
If you guess rightly, you will get: 120

If you guess wrong, wou will get: 0

select your choice

- big number
- small number

You can press Button Save and Quit to save the profile and credit and close the window.



gameform1

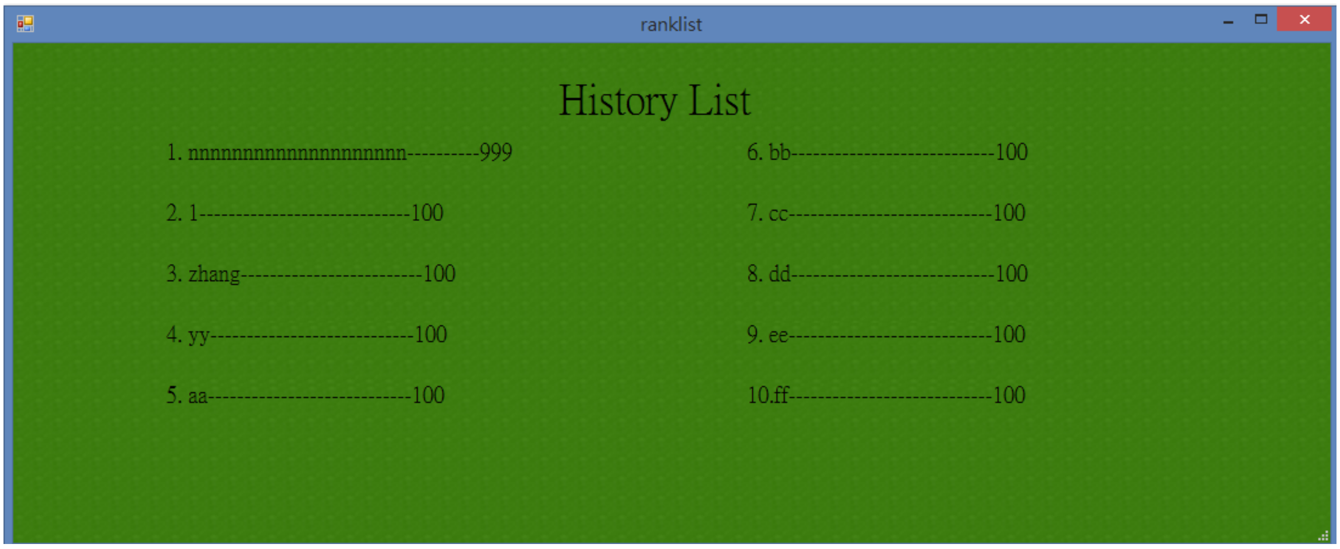
Please enter the paid score and press start to start the game

you can only choose 3 pictures to swap

paid 20

score 130

You can press Button Rank to open the History Rank List to see the top ten player of this great and interesting game!!!



History List

```
1. nnnnnnnnnnnnnnnnnnnn----------999        6. bb--------------------------100
2. 1---------------------------100          7. cc--------------------------100
3. zhang-----------------------100          8. dd--------------------------100
4. yy--------------------------100          9. ee--------------------------100
5. aa--------------------------100          10.ff--------------------------100
```