Card Game

Cgame

Yuantian Zhou

CIS 5

Winter 2019

40137

TABLE OF CONTENTS

Introduction

A card game to compare the two players' card to depend who is winner. The game will shuffle the cards each time, then the players will get the top 10 of the card pool one by one.

How the Card Game Works

```
The game compares two players' cards, the larger one wins.
```

Object of the Game

To compare two players' cards to depend who is the winner.

Rules of the Game

Card game is typically a two person game. The game is very simple:

1. Shuffle and deal the cards evenly between the two players. The players will get the top ten of cards from the card pool. Therefore, each player should have 5 cards. Jokers are not used in this game.

2. Then the game will sort their cards.

3. The game will compare player's card. The one who has the larger cards/type win the game.

4. Rule of type of cards: same color straight(12345,same color) > four and one(11112) > three and two(11122) > same color(3,5,7,8,9,same color)>straight(1,2,3,4,5,diff rent color)>three and others(333,12)>two and others(11,3,5,8) > mess(1,4,6,8,9).

How to Win the game

The one has the larger cards or larger cards group type win. If both cards are same, them the game is tied.

My Approach to the Game

Translating Game Play Rules to Programming Language

While thinking about how I was going to program this game,a questions arose:

- I don't know how to set "10" as a number of the cards, since it's a double digit. It will be missed by list the number with other numbers.

After a couple of hours of thinking, I decide to use a character B to represent 10, and create a function to translate B to 10.

Similarities to the Card Game

My War program follows the same rules of play as the card game:

- The players get the card for random.
- In some cases, the winner/loser is depend on the cards' number.

Differences from the Card Game

- Two players only take 5cards for each from the card pool.
- Not only compare the number on the cards, it also use cards group type to decide who is winner. If two are same type of group, then compare the number of the cards.

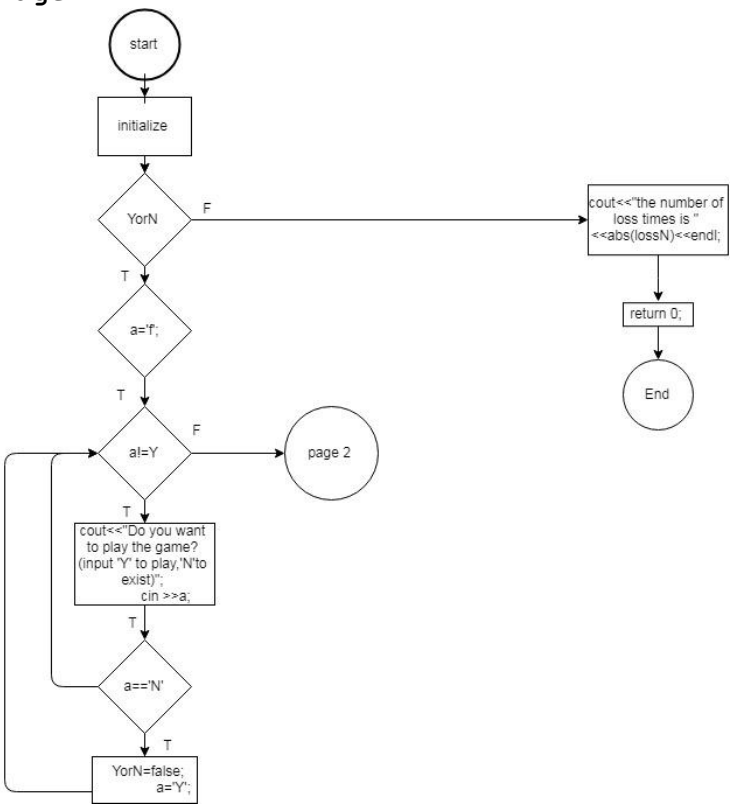
Since the face cards don't have a numeric value in the regular card game, I assigned a value to each according to their rank in the game. Therefore, the Jack is valued at 11 points, with the Queen at 12 points, and the King at 13,and the Ace is valued at 1 points.

The relationship of each number is:2<4<5<6<7<8<9<10<11<12<13<1<3.

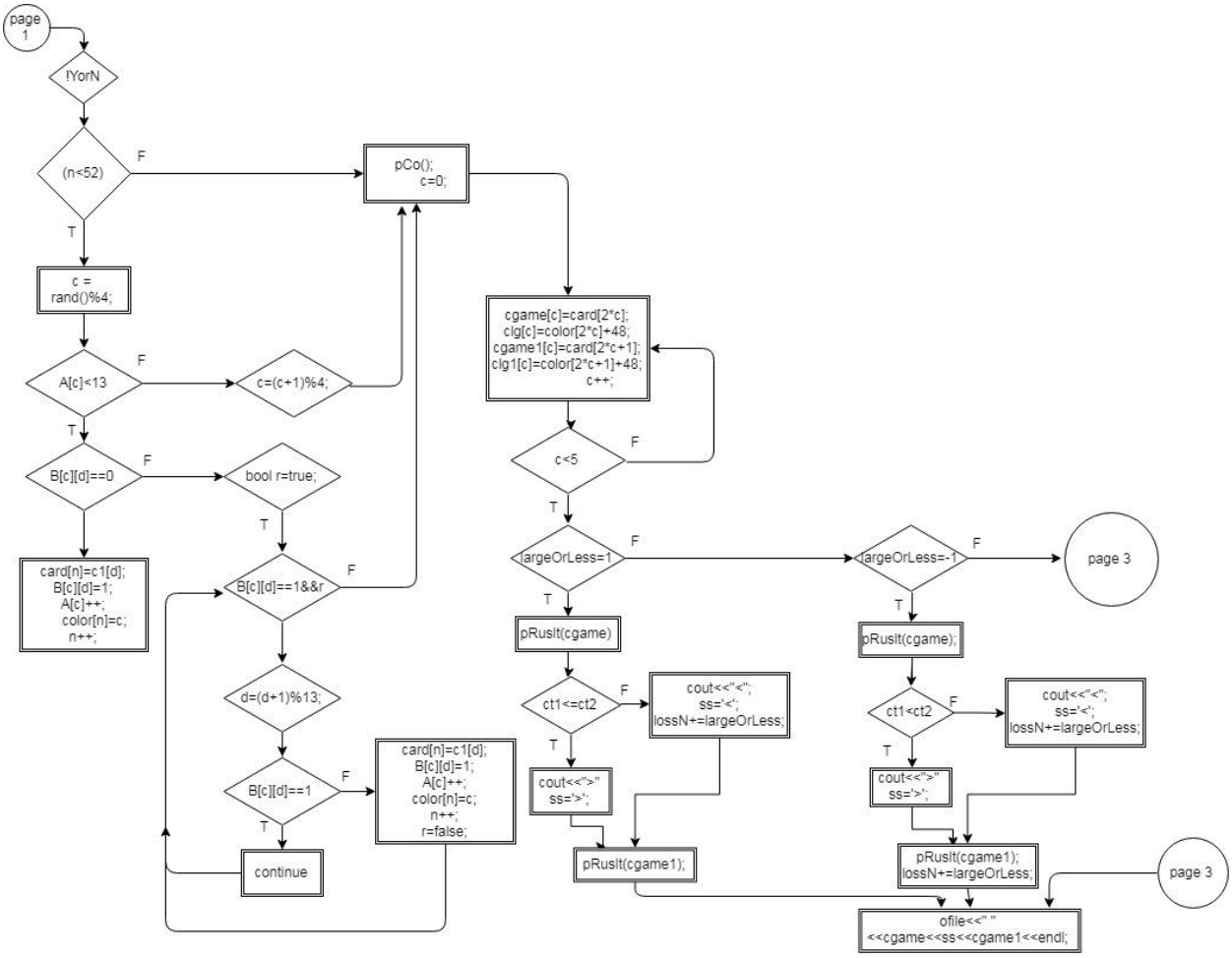
Relationship of cards group type: same color straight(12345,same color) > four and one(11112) > three and two(11122) > same color(3,5,7,8,9,same color)>straight(1,2,3,4,5,diff rent color)>three and others(333,12)>two and others(11,3,5,8) > mess(1,4,6,8,9).

Flowchart:

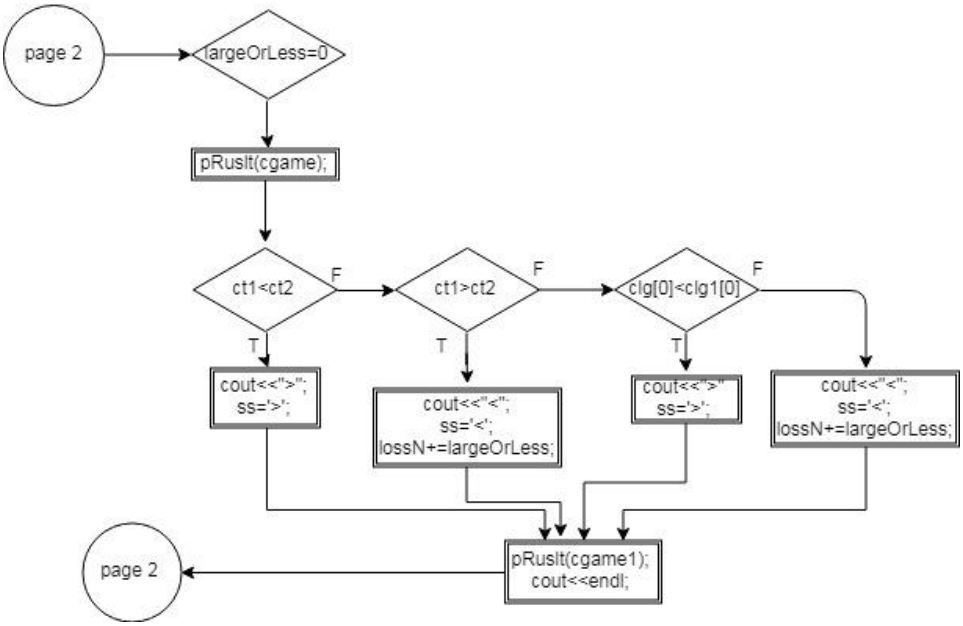
Page 1:



Page 2:


page 1
!YorN
(n<52)
F
T
pCo();
c=0;
c =
rand()%4;
A[c]<13
F
T
c=(c+1)%4;
B[c][d]==0
F
T
bool r=true;
card[n]=c1[d];
B[c][d]=1;
A[c]++;
color[n]=c;
n++;
B[c][d]==1&&r
F
T
d=(d+1)%13;
B[c][d]==1
F
T
card[n]=c1[d];
B[c][d]=1;
A[c]++;
color[n]=c;
n++;
r=false;
continue
cgame[c]=card[2*c];
clg[c]=color[2*c]+48;
cgame1[c]=card[2*c+1];
clg1[c]=color[2*c+1]+48;
c++;
c<5
F
T
largeOrLess=1
F
T
pRuslt(cgame)
ct1<=ct2
F
T
cout<<"<";
ss='<';
lossN+=largeOrLess;
cout<<">"
ss='>';
pRuslt(cgame1);
largeOrLess=-1
F
T
page 3
pRuslt(cgame);
ct1<ct2
F
T
cout<<"<";
ss='<';
lossN+=largeOrLess;
cout<<">"
ss='>';
pRuslt(cgame1);
lossN+=largeOrLess;
page 3
ofile<<" "
<<cgame<<ss<<cgame1<<endl;

Page 3:


page 2
largeOrLess=0
pRuslt(cgame);
ct1<ct2
F
T
cout<<">";
ss='>';
ct1>ct2
F
T
cout<<"<";
ss='<';
lossN+=largeOrLess;
clg[0]<clg1[0]
F
T
cout<<">"
ss='>';
cout<<"<";
ss='<';
lossN+=largeOrLess;
page 2
pRuslt(cgame1);
cout<<endl;

Pseudo

```
Initialize.
```

Ask the user for playing the game.
If no, game out;
Show the number of lose time.
If yes, continue playing.
Shuffle the 52 cards.
While n is smaller than 52, set random variable c.(0,1,2,3,):
If A[c] smaller than 13, set random variable d(0~13); else c equal to remainder of (c+1)/4;
If B[c][d] is 0;
card[n] equal to c1[d];
B[c][d] equal to 1;
A[c] equal to A[c]+1;
Color[n] equal to c;
n is equal to n+1;
Else bool r is true:
While B[c][d] equal to 1 and r:
d equal to remainder of (d+1)/13,
If B[c][d] equal to 1: continue,
Else: card[c] equal to c1[d];
B[c][d] equal to 1;
A[c] equal to A[c]+1;
color[n] equal to c;
n equal to n+1;
r is false;

Run function pCo();
c=0;
Do while c smaller than 5:
Cgame[c] equal to card[2*c],
Clg[c] equal to color[2*c]+48,
C equal to c+1;

while c<5:
Cgame[5] equal to 0,
Clg[5] equal to 0,
Run function pResult(cgame),
output a space,
Run function pResult(cgame1),
output: clg,clg1;
Run function sortCgame(cgame,clg)
Run function sortCgame(cgame1,clg1)
Run function pResult(cgame).
output a space;
Run function pResult(cgame1),
output clg,clg1;

Initialize an integer: largeOrless,
Run function lorl(cgame,cgame1,largeOrless),
Switch(largeOrless)

case largeOrless is 1:
Run function pResult(cgame),
output >;
ss is <;
Run function pResult(cgame1),

Case largeOrless is -1;
Run function pResult(cgame).
output<;
ss is <;
Run function pResult(cgame1)
loss equal to largeOrless+1;

Case largeOrless is 0;
pResult(cgame);
output =,
ss is =,
Run function pResult(cgame1);
ct1 equal to fSeq(cgame,clg);
ct2 equal to fSeq(cgame1,clg1);

Open file,
Out put cgame,ss,cgame1 to file;
Close file.
Output:the number of loss time is absolute value of lossN.
Return 0, the program is complete.

Constructs & Concepts Utilized

iostream Library

Name	Frequency	Description	Location
cout	19	Output Data	Throughout
cin	1	Input Data	57

cstdlib Library

Name	Frequency	Description	Location
srand()	1	Random # seed	Line 69
rand()	2	Generates rand #	Line 76,79

ctime Library

Name	Frequency	Description	Location
time	1	Set current time	Line 69

iomanip Library

Name	Frequency	Description	Location
setw()	4	Format final game stats	Line 245~250

string Library

Name	Frequency	Description	Location
string	1	Declare var.	Line 27

fstream

Name	Frequency	Description	Location
out.open()	1	Open file	Line 223
out.close()	1	Close file	Line 226
out	2	Write to file	Line 224,225
ofstream	1	Declare var.	Line 43

cmath Library

Name	Frequency	Description	Location
abs()	1	Neg. Score Alert Point Difference	Line 229

Data Types:

Data Types	Frequency	Location
int	5	Line 43,70~73
char	9	Line 20~26
string	1	Line 27
float	2	Line 46
ofstream	1	Line 43

Conditional Statements:

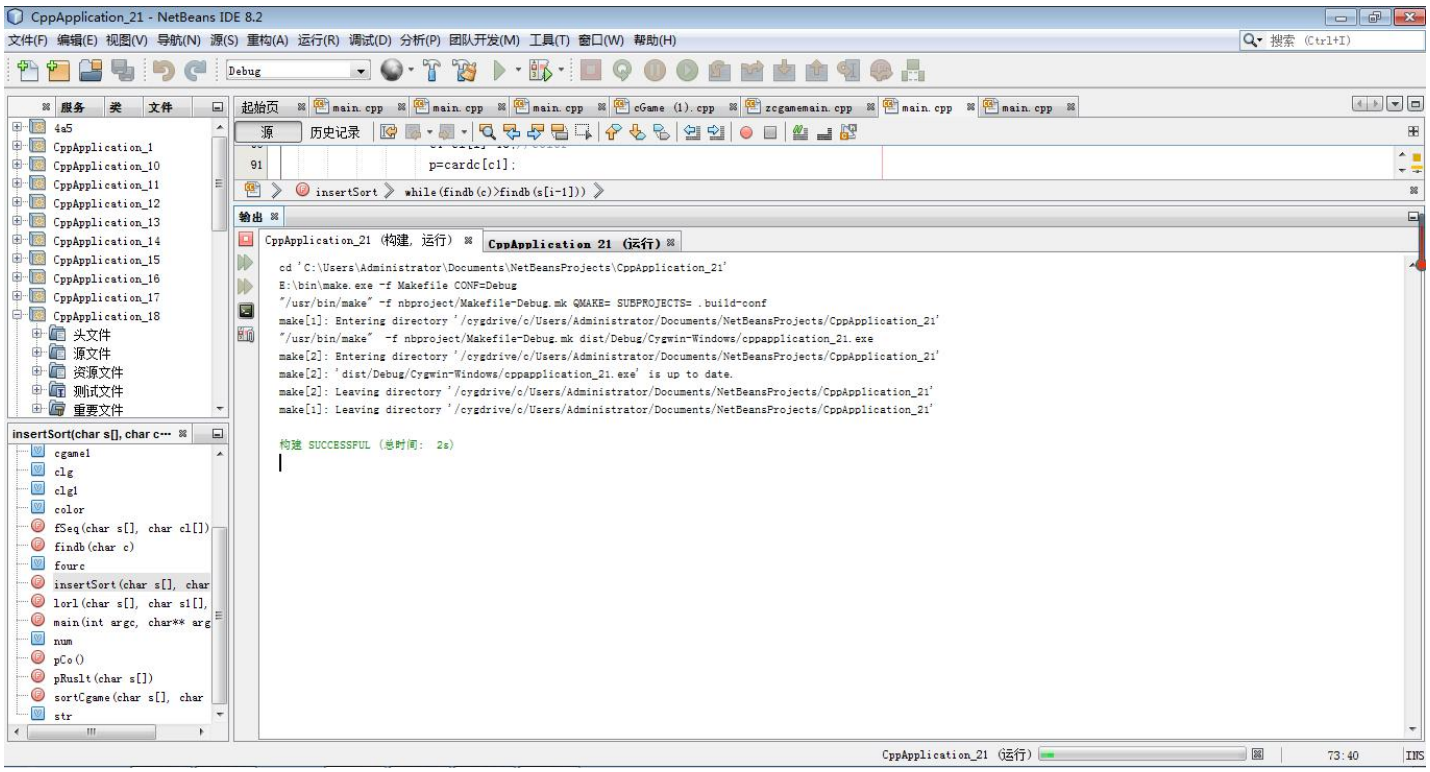
Conditional Statement	Frequency	Starting Location
if	2	Line 58,65
if/else	1	Line 80,88
if/else if	1	Line 194
switch	1	Line 146

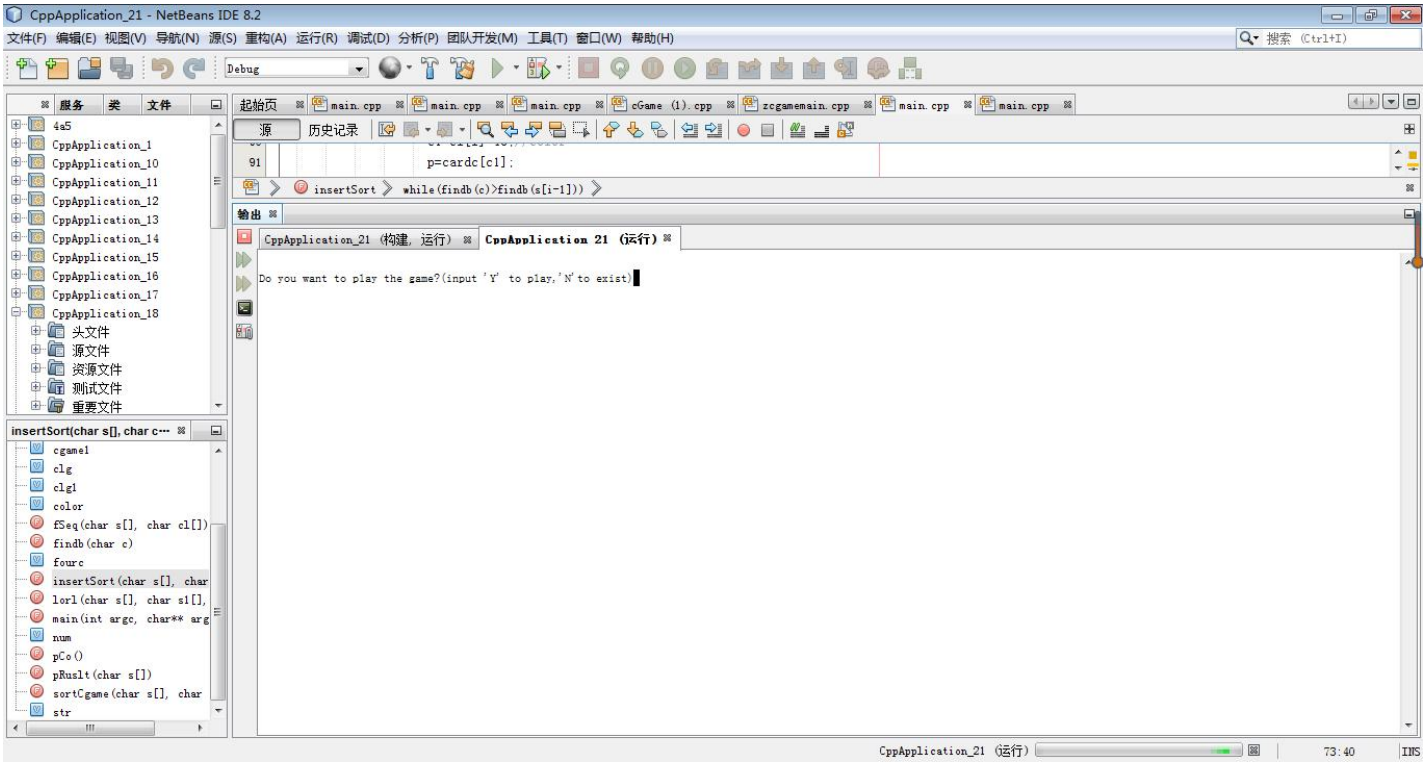
Loops:

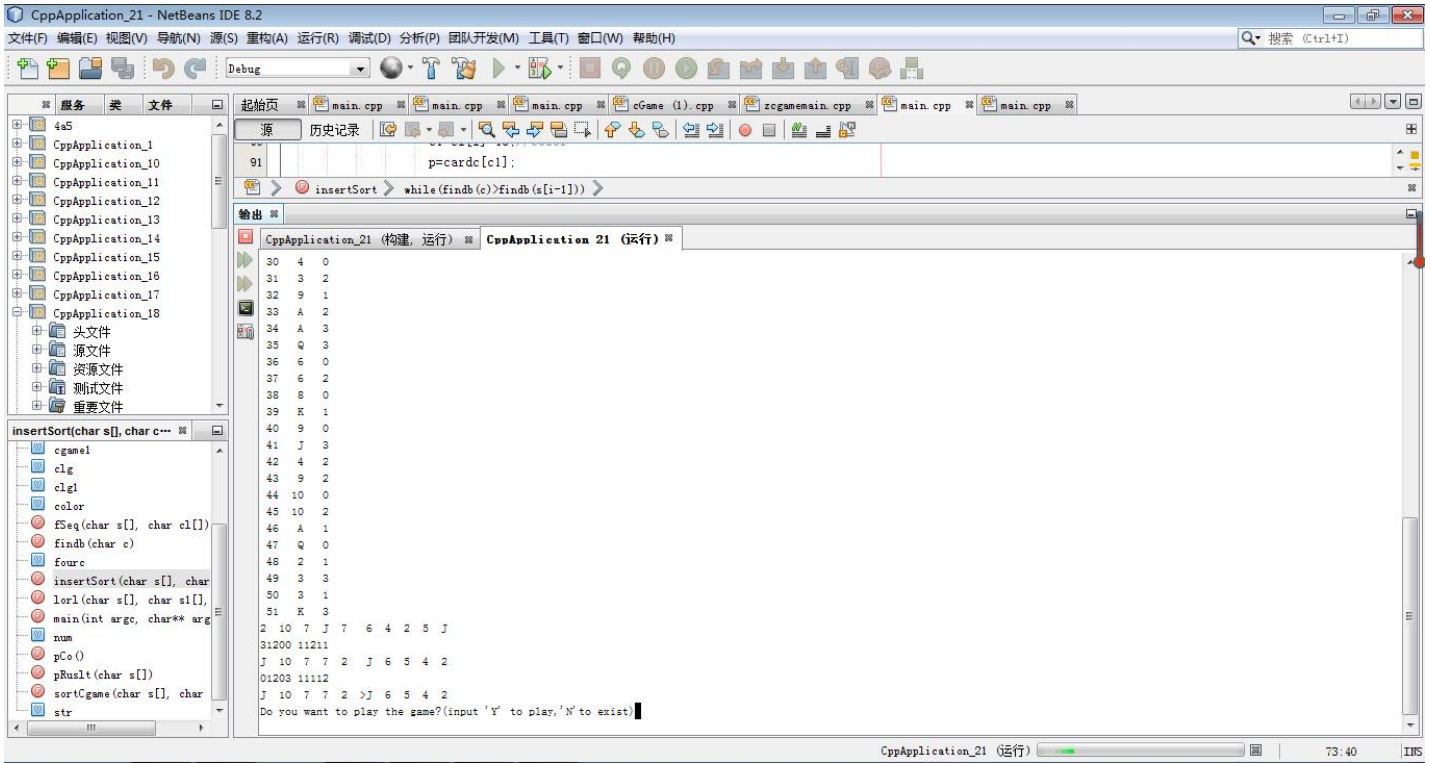
Loops	Frequency	Starting Location
for	3	Line 354,325,328
while	8	Line 320,349,301,288,274, 271,256,125
do-while	1	Line 118

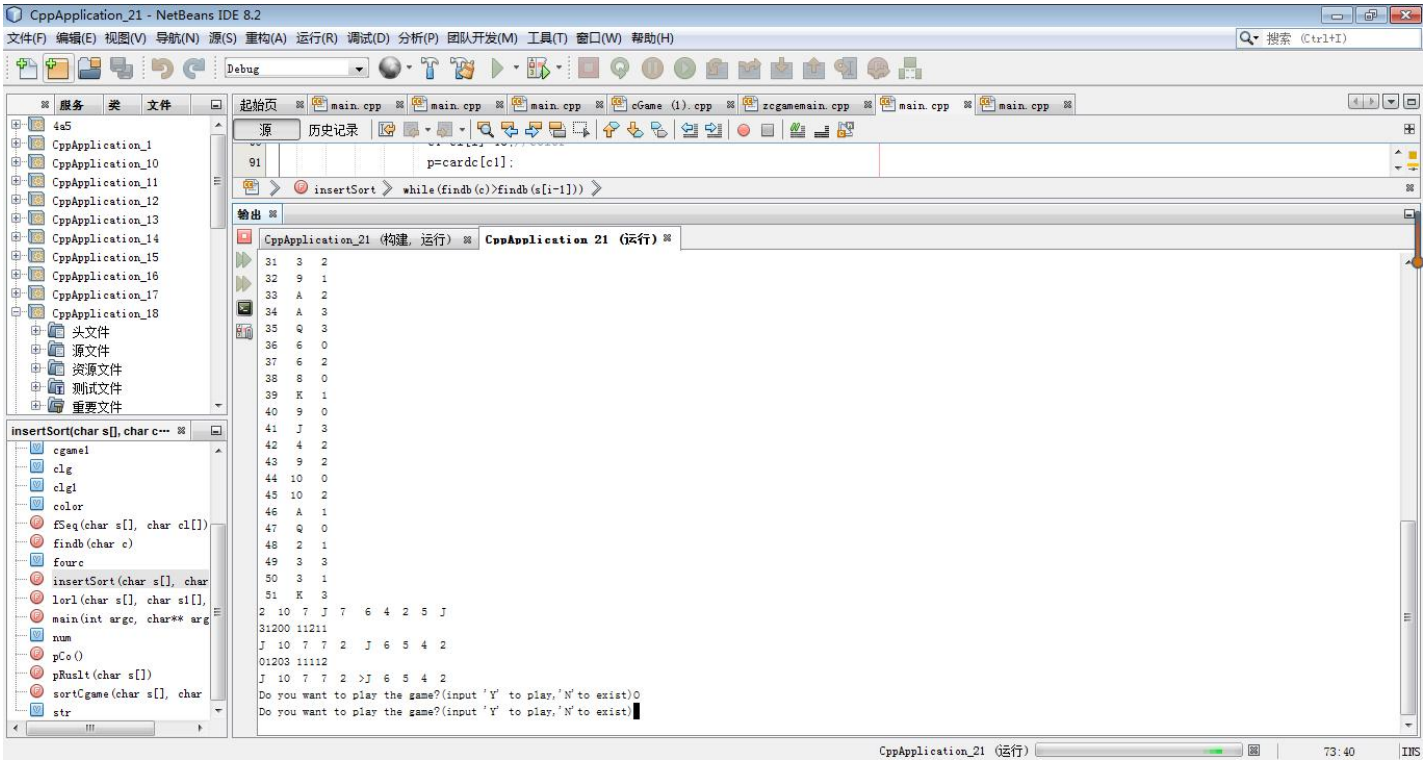
Proof of a Working Product

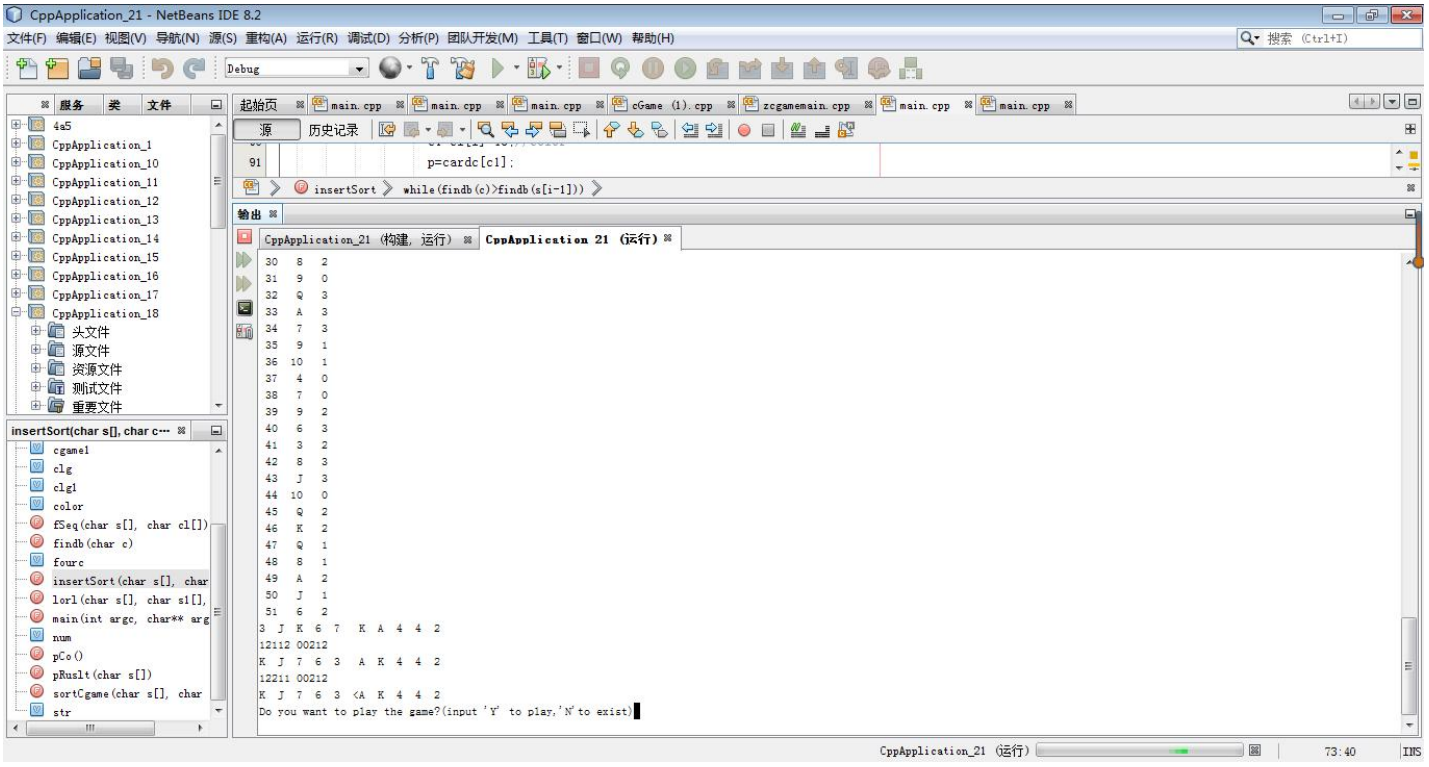
In the event, that my program does not work once it reaches Dr. Lehr, I have provided some screenshots that prove that the program did work at one time on the next few pages.

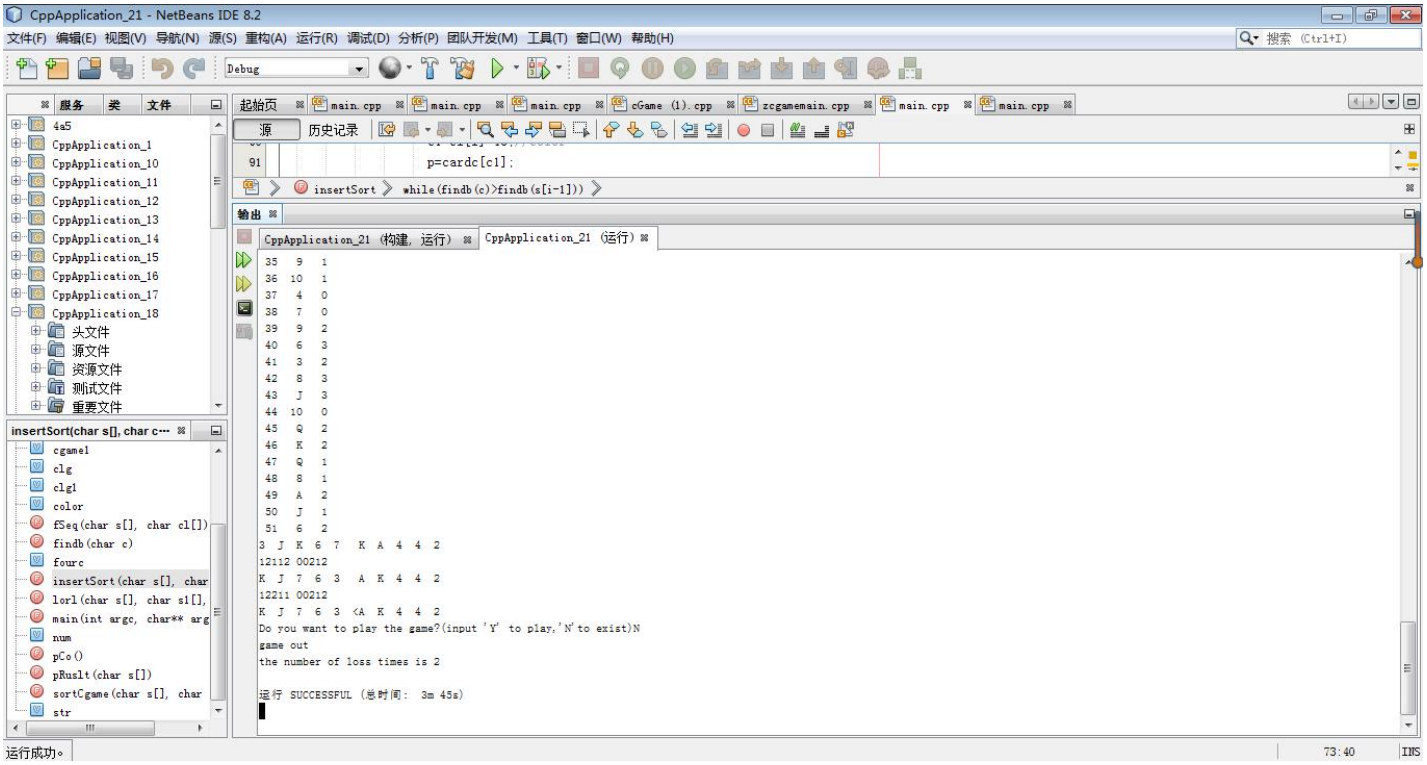



CppApplication_21 - NetBeans IDE 8.2
Do you want to play the game?(input 'Y' to play,'N' to exist)


CppApplication_21 - NetBeans IDE 8.2
Do you want to play the game?(input 'Y' to play,'N' to exist)


CppApplication_21 - NetBeans IDE 8.2
文件(F) 编辑(E) 视图(V) 导航(N) 源(S) 重构(A) 运行(R) 调试(D) 分析(P) 团队开发(M) 工具(T) 窗口(W) 帮助(H)
Debug
p=cardc[c1];
insertSort > while(findb(c)>findb(s[i-1]))
CppApplication_21 (构建, 运行)
CppApplication 21 (运行)
31 3 2
32 9 1
33 A 2
34 A 3
35 Q 3
36 6 0
37 6 2
38 8 0
39 K 1
40 9 0
41 J 3
42 4 2
43 9 2
44 10 0
45 10 2
46 A 1
47 Q 0
48 2 1
49 3 3
50 3 1
51 K 3
2 10 7 J 7 6 4 2 5 J
31200 11211
J 10 7 7 2 J 6 5 4 2
01203 11112
J 10 7 7 2 >J 6 5 4 2
Do you want to play the game?(input 'Y' to play,'N' to exist)0
Do you want to play the game?(input 'Y' to play,'N' to exist)
CppApplication_21 (运行)
73:40
INS


CppApplication_21 - NetBeans IDE 8.2
文件(F) 编辑(E) 视图(V) 导航(N) 源(S) 重构(A) 运行(R) 调试(D) 分析(P) 团队开发(M) 工具(T) 窗口(W) 帮助(H)
Debug
p=cardc[c1];
insertSort > while(findb(c)>findb(s[i-1]))
CppApplication_21 (构建, 运行)
CppApplication 21 (运行)
30 8 2
31 9 0
32 Q 3
33 A 3
34 7 3
35 9 1
36 10 1
37 4 0
38 7 0
39 9 2
40 6 3
41 3 2
42 8 3
43 J 3
44 10 0
45 Q 2
46 K 2
47 Q 1
48 8 1
49 A 2
50 J 1
51 6 2
3 J K 6 7 K A 4 4 2
12112 00212
K J 7 6 3 A K 4 4 2
12211 00212
K J 7 6 3 <A K 4 4 2
Do you want to play the game?(input 'Y' to play,'N' to exist)
CppApplication_21 (运行)
73:40
INS



References

1. Dr. Lehr's Lectures & Lab
2. "Starting Out with C++: From Control Structures through Objects" Gaddis, Savitch .(Textbook)

Program

```
/*
 * File:   main.cpp
 * Author: Yuantian Zhou
 *
 * Created on 2019.1.30
 * Two players select 5 cards for each for random,
 * the one who has large number or large group of card wins
 * compare the cards one by one
 */
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <iomanip>
#include <fstream>
```

```
#include <cmath>
#include <string>
#include <vector>
using namespace std;
  char card[52];//total 52 cards;
  char color[52]="0123" ;//4 classes of cards;
  char cgame[6];//player1's cards;
  char clg[6];//player1's cards' color;
  char cgame1[6];//player2's cards;
  char clg1[6];//player2's color;
  char c1[14] ="23456789BJQKA";//cards' numbers
  string str[3]={"a","b","c"};
 char fourc[4][6];
 char cardc[4][6];
  int num[14];
  template <class T>
 void Pvector(const vector<T> & v);
void pCo(); //print the card pool
void pRuslt(char s[]);//print the player's card
int findb(char c);//find the index to compare
 void insertSort(char s[],char c);
    int  fSeq(char s[],char cl[]);
void sortCgame(char s[],char st[]);//sort the card pool
bool  lorl(char s[],char s1[],int &n); //compare players cards
void sortCgame(char s[]);
int main(int argc, char** argv) {
    char ss;//for calculate the loss times
        ofstream ofile;
    char fname[20]="result.txt";//file type:txt
    char a;
    float lossN=0;//number of times of loss
    int d[5]={1,2,3,4,5};
    vector<int> v(d,d+1);
        Pvector(v);
    bool YorN=true;
    while(YorN)
    {
        a='f';
        while(a!='Y')//valid input
        {
            cout<<"Do you want to play the game?(input 'Y' to play,'N'to exist)";
            cin >>a;
            if(a=='N')
            {
                cout<<"game out"<<endl;
                YorN=false;
                a='Y';
            }
        }
        if(!YorN)
        {
            continue;
        }
        srand(static_cast<unsigned int>(time(0)));
     int A[4]={0};
     int B[4][13]={0};
     int n=0;
     int c,d;
     while(n<52)
     {
         c = rand()%4;//0 is spade,1 is heart, 2 is club,3 is diamand
           if (A[c]<13)
         {
             d = rand()%13;
             if (B[c][d]==0)
```

```
            {
                card[n]=c1[d];
                B[c][d]=1;
        A[c]++;
                color[n]=c;
                        n++;
            }
            else
            {
                bool r=true;
                while(B[c][d]==1&&r)
       {
              d=(d+1)%13;
              if (B[c][d]==1)
                    continue;
                       else
                       {
                    card[n]=c1[d];
                    B[c][d]=1;
                                A[c]++;
                    color[n]=c;
                    n++;
                    r=false;
                         }
                };

            }
         }
         else
         {
             c=(c+1)%4;

         }
     }

     pCo();
          c=0;
     do
     {
         cgame[c]=card[2*c];
        clg[c]=color[2*c]+48;
        cgame1[c]=card[2*c+1];
        clg1[c]=color[2*c+1]+48;
                c++;
     }while(c<5);
         //test straight draw,and same color straight
       cgame[5]=0;
clg[5]=0;
pRuslt(cgame);
cout<<" ";
pRuslt(cgame1);
cout<<endl;
cout<<clg<<" "<<clg1<<endl;
sortCgame(cgame,clg);
sortCgame(cgame1,clg1);
pRuslt(cgame);
cout<<" ";
pRuslt(cgame1);
cout<<endl;
cout<<clg<<" "<<clg1<<endl;
int largeOrLess;
lorl(cgame,cgame1,largeOrLess);
int ct1,ct2;
ct1=fSeq(cgame,clg);//ct1,ct2 repersent the cards type
ct2=fSeq(cgame1,clg1);
```

```
switch(largeOrLess){
    case 1:
        {
      pRuslt(cgame);
      if(ct1<=ct2)
      {
             cout<<">";// win
             ss='>';
      }
      else
      {//lose
             cout<<"<";
             ss='<';
             lossN+=largeOrLess;
      }

      pRuslt(cgame1);
      cout<<endl;
                break;
        }
    case -1:
        {
                    pRuslt(cgame);

                    if(ct1<ct2)
                    {
                           cout<<">";// win
                           ss='>';
                    }
                    else
                    {//lose
                           cout<<"<";
                           ss='<';
                           lossN+=largeOrLess;
                    }
                           pRuslt(cgame1);
                           cout<<endl;
                //lossN+=largeOrLess;
                break;
             }
    case 0: //
        {
                    pRuslt(cgame);
                    if(ct1<ct2)
                    {
                           cout<<">";// win
                           ss='>';
                    }
                    else if(ct1>ct2)
                    {//lose
                           cout<<"<";
                           ss='<';
                           lossN+=largeOrLess;
                    }
                    else
                    {
                           if(clg[0]<clg1[0])
                           {
                                  cout<<">";// win
                                  ss='>';

                           }
                           else
                           {
                                  cout<<"<";
```

```
                                        ss='<';
                                        lossN+=largeOrLess;

                                    }


                                }
                                        pRuslt(cgame1);
                                        cout<<endl;

                        }
                }
            ofile.open(fname,ios::app);
                ofile<<" "<<cgame<<ss<<cgame1<<endl;
                ofile<<endl;
                ofile.close();

        }
        cout<<"the number of loss times is "<<abs(lossN)<<endl;
        exit(0);
}
template <class T>
 void Pvector(const vector<T> & v)
{
    typename vector<T>::const_iterator i;
    for(i=v.begin();i!=v.end();++i)
        cout<<endl;
 }

void pCo() //print the card pool
{
 cout<<endl;
    for(int c=0;c<52;c++)
     {
       cout<<setw(3)<<c<<" ";
         if(card[c]=='B') //B is symbol of "10"
             cout<<setw(3)<<  10 << " "; //change "B" to "10";
         else
             cout<<setw(3)<< card[c]<<" ";
         cout<<setw(3)<<char(color[c]+'0')<<endl;
     }
}
void pRuslt(char s[])//print the player's card
{
 int i=0,len=0;
 while(s[i++])
        len++;
 for(i=0;i<len;i++)
            (s[i]=='B')?cout<<10<<"  ":cout<<s[i]<<"  ";
 }
int findb(char c)//find the index to compare
{
 for(int i=0;i<13;i++)
        if(c1[i]==c)
              return i;
}
void insertSort(char s[],char c)
  {//binary sort
   char temp;
   int i=0,len=0;
   while(s[i++])
          len++;
   i=len;
   while(findb(c)>findb(s[i-1]))
   {
```

```
        s[i]=s[i-1];
        i--;
   }
   s[i]=c;
  }
int  fSeq(char s[],char cl[])
  {//determind the type of cards group, such as straight.
   //same color straight,four and one,three and two,same color,straight,three,two,mess
   int i=0,c1;
   int k[4]={0},delta=1;
   int ctype;//
   char c,*p;
   while(i<5)
   {
        c=s[i];//card
        c1=cl[i]-48;//color
        p=cardc[c1];
        insertSort(p,c);
        k[c1]++;
        i++;
   }
   if(k[0]==5||k[1]==5 || k[2]==5 || k[5]==5)
   {
        ctype=1;//assume its same color straight
        i=0;
        while(i<4 && delta==1 && ctype==1)
        {
             delta=findb(s[i])-findb(s[i+1]);
             if(delta!=1)
                  ctype=4;//is same color straight
             i++;
        }

   }
   else
      ctype=2;//if is not straight or same color straight repeat test
      return ctype;


  }
void sortCgame(char s[],char st[])//sort the card pool
{//selection sort
 int i=0,j,k,len=0;
 char temp;
 while(s[i])
 {
       len++;
       i++;
 }
 for(i=0;i<len-1;i++)
 {
       k=i;
       for(j=i+1;j<len;j++)
       {
            if(findb(s[k])<findb(s[j]))
                 k=j;
       }
       if(k!=i)
       {
            temp=s[i];
            s[i]=s[k];
            s[k]=temp;
            temp=st[i];
            st[i]=st[k];
            st[k]=temp;
```

```
        }

 }
}
bool  lorl(char s[],char s1[],int &n) //compare players cards
{
 int i=0,len=0;
 char temp;
 while(s[i])
 {
        len++;
        i++;
 }
 for(i=0;i<len;i++)
 {
        if(findb(s[i])<findb(s1[i]))
                 {
                     n= -1;
                     return true;
                 }
        if (findb(s[i])>findb(s1[i]))
                 {
               n= 1;
                           return true;
                 }
 }
 n= 0;
         return false;
}
void sortCgame(char s[])
{
    cout<<" ";
}
```