Project 1

<Black Jack (21 Point)>

CSC 5 - 48102
Name:Weikang Du
Date: 10/28/16

Introduction

Title : Black Jack (21 Point)

Blackjack, also known as twenty-one, is the most widely played casino banking game in the world.[1] Blackjack is a comparing card game between a player and dealer, meaning players compete against the dealer but not against other players. It is played with one or more decks of 52 cards. The objective of the game is to beat the dealer in one of the following ways:

Get 21 points on the player's first two cards (called a "blackjack" or "natural"), without a dealer blackjack;
Reach a final score higher than the dealer without exceeding 21; or
Let the dealer draw additional cards until his or her hand exceeds 21.

Actually, I changed some rule of this game:

There 52 cards, and each round has 4 times to send card, each time for 1 card.

I made the game to Player vs. Computer. There are two mode for player:

1. Normal Player: Player is able to know how many points in hand, but he/she could not know the next card number, therefore, this mode is hard one and player want to get closer 21 but not exceeds 21 as possible.

2. Super Player: Player also can know how many points in hand, and he/she has power to see the number of next card and decide to choose or deny.

Summary

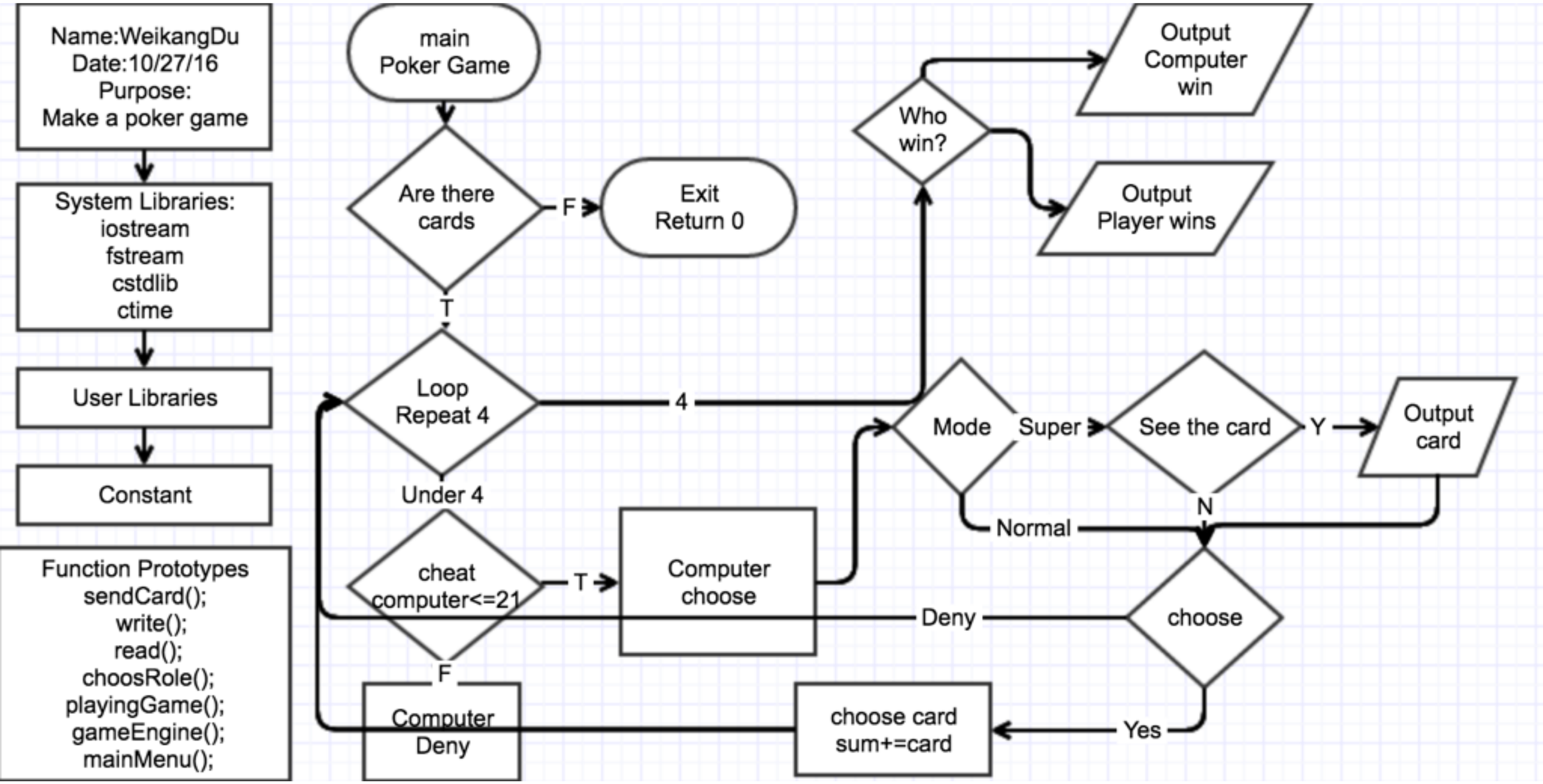
Project Size: 200+ lines

The number of variables: about 16

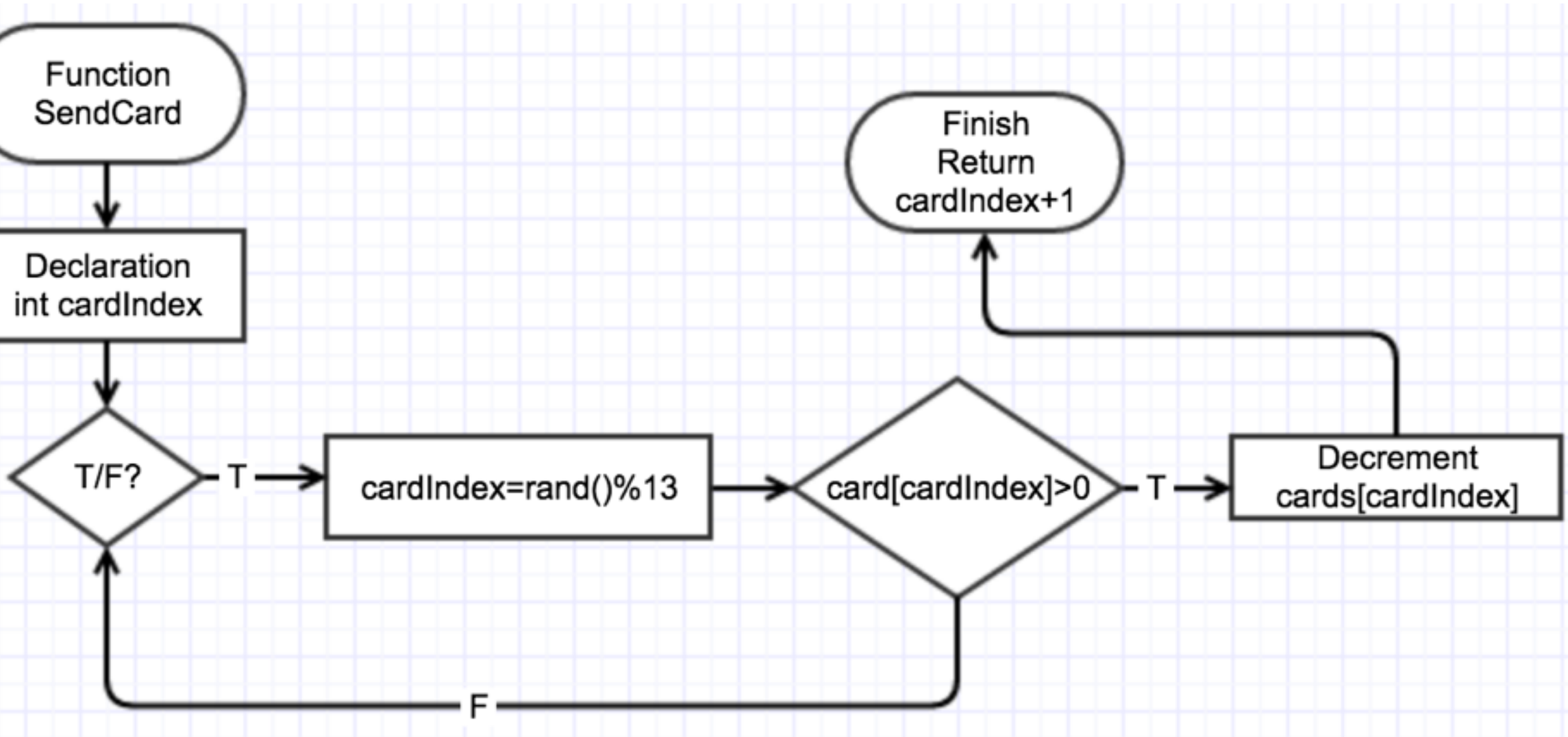
I made it around one week.
It was hard to make the partial of computer, therefore,
the computer can predict whether it will exceed 21. So
the computer will never exceed 21, however, the player
can not predict that.
And also, I used some array I learn from internet and book.

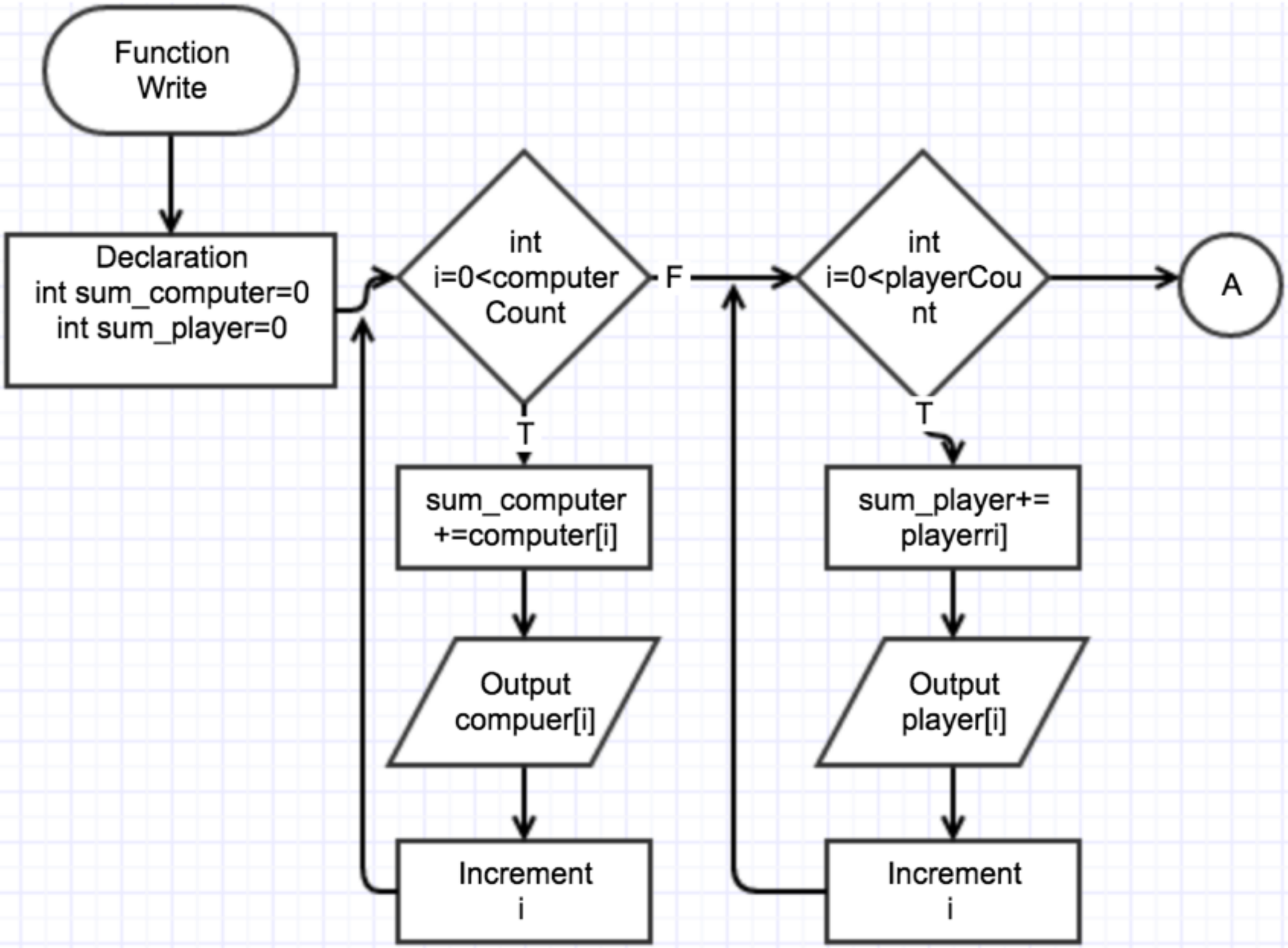
Actually, I do not know how to make a flowchart
with function, so I made the outline of main program
and then I made some important function in this write up.

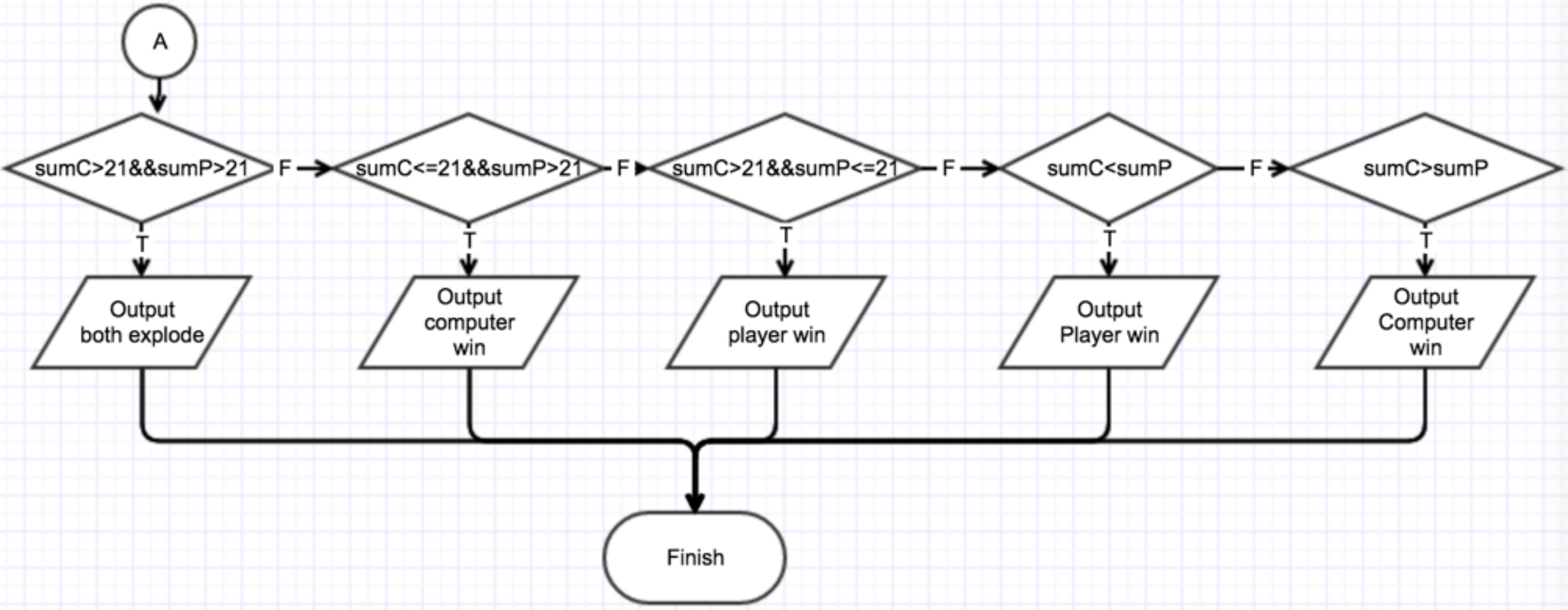
Flowchart:


Name:WeikangDu
Date:10/27/16
Purpose:
Make a poker game
System Libraries:
iostream
fstream
cstdlib
ctime
User Libraries
Constant
Function Prototypes
sendCard();
write();
read();
choosRole();
playingGame();
gameEngine();
mainMenu();
main
Poker Game
Are there
cards
F
Exit
Return 0
T
Loop
Repeat 4
4
Under 4
cheat
computer<=21
T
Computer
choose
F
Computer
Deny
Who
win?
Output
Computer
win
Output
Player wins
Mode
Super
See the card
Y
Output
card
Normal
N
choose
Deny
Yes
choose card
sum+=card

SendCard


Function
SendCard
Declaration
int cardIndex
T/F?
T
cardIndex=rand()%13
card[cardIndex]>0
T
Decrement
cards[cardIndex]
Finish
Return
cardIndex+1
F


Function
Write
Declaration
int sum_computer=0
int sum_player=0
int
i=0<computer
Count
F
T
sum_computer
+=computer[i]
Output
compuer[i]
Increment
i
int
i=0<playerCou
nt
A
T
sum_player+=
playerri]
Output
player[i]
Increment
i


A
sumC>21&&sumP>21
F
sumC<=21&&sumP>21
F
sumC>21&&sumP<=21
F
sumC<sumP
F
sumC>sumP
T
T
T
T
T
Output
both explode
Output
computer
win
Output
player win
Output
Player win
Output
Computer
win
Finish

SetIcon	Describtion
card[13]={4,4,4,4,4,4,4,4,4,4,4,4,4}	There are 13 kinds of card and each kind for 4. so total card number is 52.
computerCount	The number of times the computer got the card
playerCount	The number of times the player got the card
rounds	The counter for round
cardIndex	Determine the each kind of card
flag	Make sure whether card is empty
sum_computer	The total points in hand of computer
sum_player	The total points in hand of player
tmd_card	Temporary card random number
choice	Choose card or Deny
role	The game mode

Program

```
//Function prototypes
int sendCard();
void write();
void read();
int choosRole();
void playingGame();
void gameEngine();
void mainMenu();
bool cardIsempty();
//Declaration the Variables
int cards[13]={ 4,4,4,4,4,4,4,4,4,4,4,4,4};
//The 13 kinds of poker and each for 4, no joker
int computer[24];
int computerCount; //The number of computer get card
int player[24];
int playerCount;   //The number of player get card
int rounds=0;      //Set the round
```

```
//The function for send card
int sendCard()
{   int cardIndex;
    while (true)
    {   //Produce a random number
        cardIndex = rand() % 13;
        //Determine whether the same point card exits
        if (cards[cardIndex] > 0)
        {
            cards[cardIndex]--;
            break;
        }
    }
    return (cardIndex + 1);
}
//Determine the total of cards
bool cardIsempty()
{
    bool flag=true;
    for (int i=0; i<13; i++)
    {
        if (cards[i]>0)
        {flag=false;
            break;}
    }
    return flag;
}
```

```
//Function for the output in file
void write(ofstream &outfile, int round)
{
    if (!outfile)
    {
        cout<<"Open file failed!"<<endl;
    }
    else
    {
        int sum_computer=0;
        int sum_player=0;
//Output the round number
 outfile<<"*****round"<<round<<"******\n";
 outfile<<"Computer:\t";
//Output the card number each time for computer
 for (int i=0; i<computerCount; i++)
 {
     sum_computer+=computer[i];
     outfile<<computer[i]<<"\t";
 }
 outfile<<"\n";
 outfile<<"Player  :\t";
```

```
//Output the card number each time for player
        for (int i=0; i<playerCount; i++)
        {
            sum_player+=player[i];
            outfile<<player[i]<<"\t";
        }
        outfile<<"\n";
        outfile<<"results:\t";
        //Determine the result and display it
        if (sum_computer>21 && sum_player>21)
        {
            outfile<<"Computer and Player are both explode!\n";
        }
        else if (sum_computer<=21 && sum_player>21)
        {
            outfile<<"Computer wins!\n";
        }
        else if (sum_computer>21 && sum_player<=21)
        {
            outfile<<"Player wins!\n";
        }
        else if (sum_computer<sum_player)
        {
            outfile<<"Player wins!\n";
        }
        else if (sum_computer>sum_player)
        {
            outfile<<"Computer wins!\n";
        }
    }
}
```

```
//Function for read and display it
void read()
{
    ifstream infile;
    infile.open("records.txt");
    if (!infile)
    {
        cout<<"Open file failed!"<<endl;
    }
    else
    {
        string str;
        while(!infile.eof())
        {
            getline(infile, str);
            if(str=="")
            {
                break;
            }
            else
            {
                cout<<str<< endl;
            }
        }
        infile.close();
    }
}
```

```
int choosRole()
{
    int choice;
    cout<<"Welcome to play the Black Jack Game (21 Points Game)"<<endl;
    cout<<"Please enter in a number to choose the mode:"<<endl;
    cout<<"1.Normal Player ( NORMAL )"<<endl;
    cout<<"2.Super  Player (  EASY  )"<<endl;
    cin>>choice;
    return choice;
}

//Function for the playing
void playingGame(ofstream &outfile)
{
int tmp_card; //Temporary card number
char choice;
int card_count=0;
int sum_computer,sum_player; //The total number in hand for both
int role;
role=choosRole();
cout<<"** Start the new round."<<endl;

srand((unsigned)time(NULL));
```

```
  while (!cardIsempty())
{
    card_count=0;
    sum_computer=0;
    sum_player=0;

    while (card_count<4)
    {
        cout << "** Send computer the card.\n";
        tmp_card=sendCard();
        if (sum_computer+tmp_card<=21)
        {
            sum_computer+=tmp_card;
            computer[computerCount] = tmp_card;
            computerCount++;
    }
    cout<<"** Send player the card.\n";
    tmp_card=sendCard();

    cout<<"** The sum of cards in hand is: "<<sum_player<<endl;
    if(role==2)
    {
        cout<<"Enter Y to see the card, or any key to deny this:"<<endl;
        cin>>choice;
        if(choice=='Y'||choice=='y')
        {
            cout<<"** The card is "<<tmp_card<<endl;
        }
    }
```

```
cout << "Enter Y to choose the card, or any key to deny the card:";
            cin >> choice;
            if (choice=='Y'||choice=='y')
            {
                sum_player+=tmp_card;
                player[playerCount]=tmp_card;
                playerCount++;
            }
            card_count++;
        }
        rounds++;
        write(outfile,rounds);
        cout << "This round ends."<<endl;
        cout << "Enter Y to continue play, or Any key to stop the game and see the result:";
        cin >> choice;

        if (choice=='Y'||choice=='y')
        {
        computerCount=0;
        playerCount=0;

        for(int j=0; j<24; j++)
        {
                computer[j]=0;
                player[j]=0;
        }
        for(int j=0; j<13; j++)
                cards[j]=4;
                card_count=0;
            continue;
        }
```

```
else
        {
            break;
        }
    }
}

void gameEngine()
{
    ofstream outfile;
    outfile.open("records.txt");
    if (!outfile)
    {
        cout<<"Open file failed!"<<endl;
    }
    else
    {
        playingGame(outfile);
        outfile.close();
        read();
    }
}
```

```
void mainMenu()
{

}

int main()
{
    gameEngine();
}
```