Project 2

<Othello>

Name: Xiaojun He

Class: CSC-17A 48096

Date: 12/07/2015

Contents

1. Introduction(From wiki):

The game *Reversi* was invented in 1883 by either of two Englishmen (each claiming the other a fraud), Lewis Waterman[1] or John W. Mollett (or perhaps earlier by someone else entirely), and gained considerable popularity in England at the end of the nineteenth century.[2] The game's first reliable mention is in the August twenty-first 1886 edition of *The Saturday Review*. Later mention includes an 1895 article in *The New York Times*: "Reversi is something like Go Bang, and is played with 64 pieces."[3] In 1893, the well-known German games publisher Ravensburger started producing the game as one of its first titles. Two 18th-century continental European books dealing with a game that may or may not be Reversi are mentioned on page fourteen of the Spring 1989 *Othello Quarterly*, and there has been speculation, so far without documentation, that the game has even more ancient origins.

Each of the disks' two sides corresponds to one player; they are referred to here as *light* and *dark* after the sides of *Othello* pieces, but any counters with distinctive faces are suitable. The game may for example be played with a chessboard and Scrabble pieces, with one player *letters* and the other *backs*.

The historical version of Reversi starts with an empty board, and the first two moves by each player are in the four central squares of the board. The players place their disks alternately with their color facing up and no captures are made. A players may choose to not play both pieces on the same diagonal, different from the standard *Othello* opening. It is also possible to play variants of Reversi and *Othello* wherein the second player's second move may or must flip one of the opposite-colored disks (as variants closest to the normal games).

For the specific game of *Othello* (as technically differing from the historical Reversi), the rules state that the game begins with four disks placed in a square in the middle of the grid, two facing white side up, two pieces with the dark side up, with same-colored disks on a diagonal with each other. Convention has initial board position such that the disks with dark side up are to the north-east and south-west (from both players' perspectives), though this is only marginally meaningful to play (where opening memorization is an issue, some players may benefit from consistency on this). If the disks with dark side up are to the north-west and south-east, the board may be rotated by 90° clockwise or counterclockwise. The dark player moves first.

2. Summary

Total Line of Code	1000+
Variable	-
Function	-
Comment Line	-

Othello is the very fun game, the use every select I should check the select is ok or not. And every select will change a lot thing from the checkerboard. I use the Othello class to get all the function, include the check, the change, the jungle, and check win. Also I make the AI for the game, it have two choice for the player, one is person with person and the other is person play with AI.

3. Problem during coding

    a) When the user select his choices, there are a lot thing change from the checkerboard. When the black player select, I should change all the white dis between two black dis to black, and save it to the checkerboard.

    b) For the game Othello, the choices is not anywhere, I should write the function to check the choices that the player make is ok or not. If yes, then go ahead, of not, then ask the user make the choices again.

4. System Libraries

#include<iostream>

#include<algorithm>

#include<windows.h>

#include<cstring>

#include<ctime>

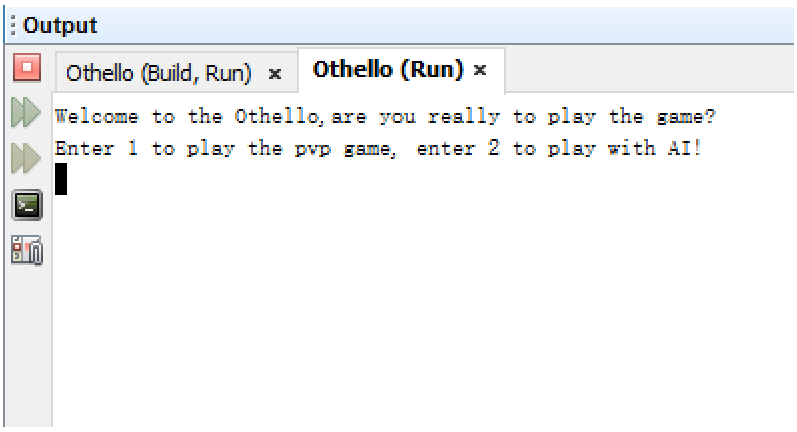
#include<cmath>

#include<cstdio>

Othello.h

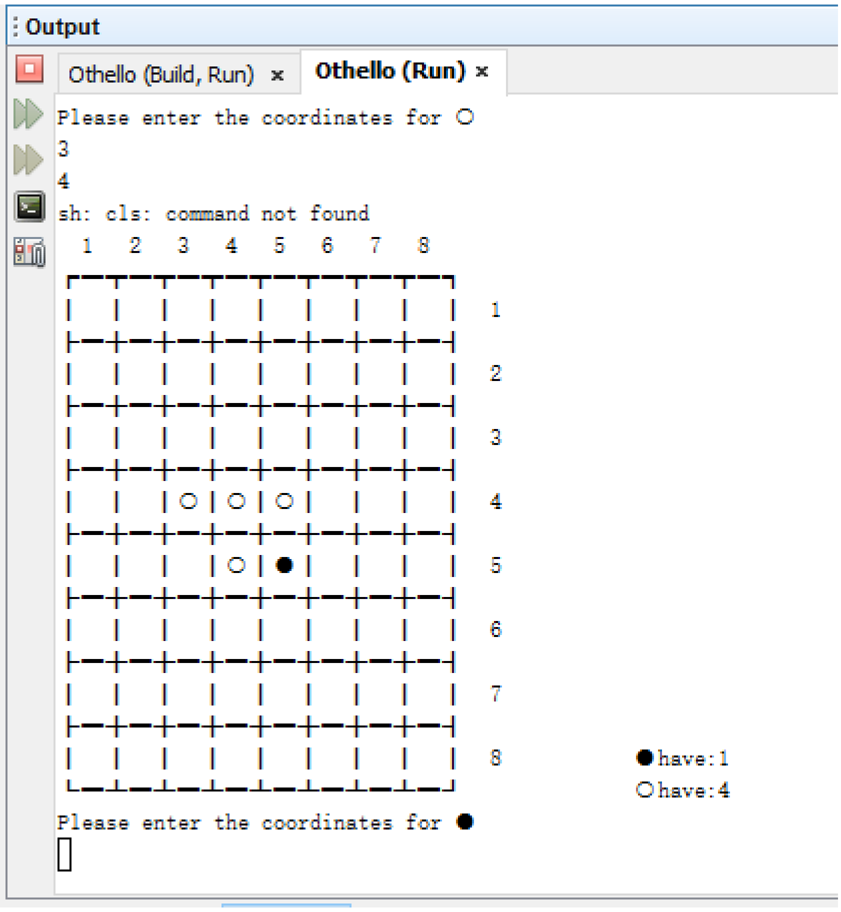
5. Screen shot

1) Ask the user input the number to choices play with AI or play with person



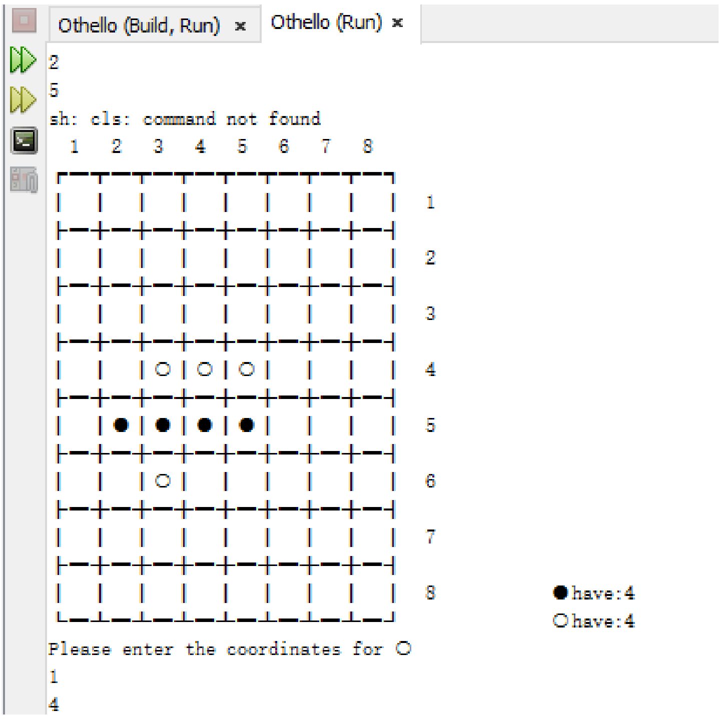
```
Welcome to the Othello, are you really to play the game?
Enter 1 to play the pvp game,  enter 2 to play with AI!
```

2) If choices 1 ask the user to enter the coordinates for while



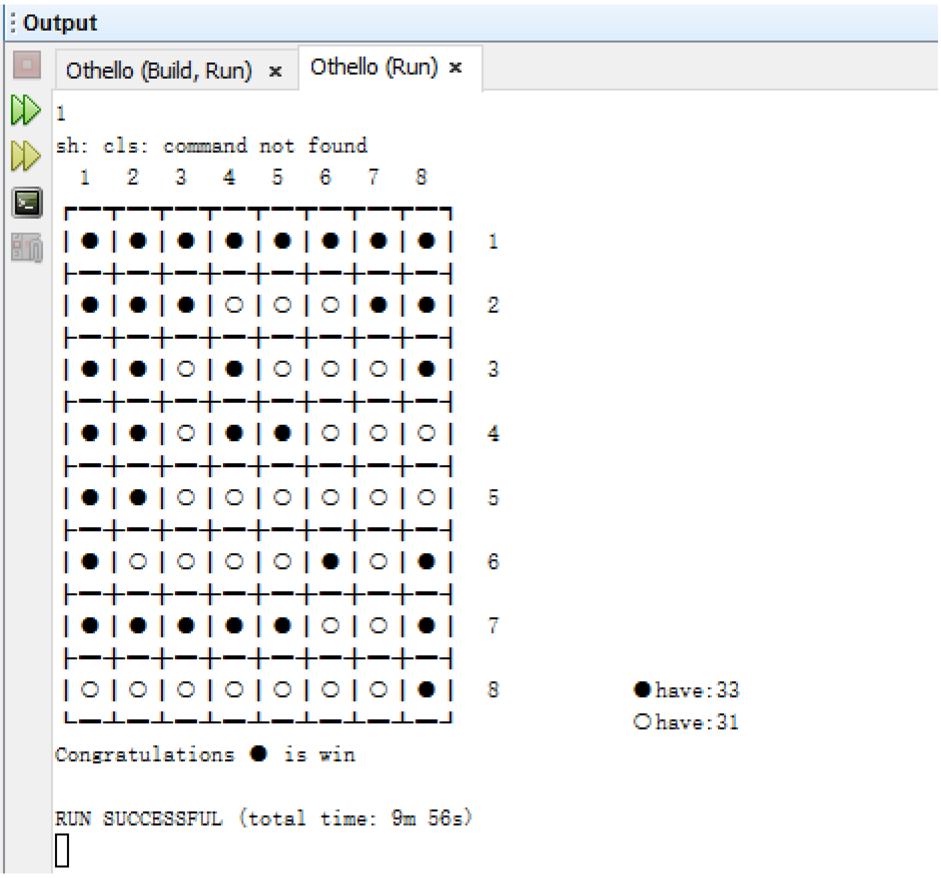
```
Please enter the coordinates for ○
3
4
sh: cls: command not found
  1   2   3   4   5   6   7   8
┌─┬─┬─┬─┬─┬─┬─┬─┐
│   │   │   │   │   │   │   │   │   1
├─┼─┼─┼─┼─┼─┼─┼─┤
│   │   │   │   │   │   │   │   │   2
├─┼─┼─┼─┼─┼─┼─┼─┤
│   │   │   │   │   │   │   │   │   3
├─┼─┼─┼─┼─┼─┼─┼─┤
│   │   │ ○ │ ○ │ ○ │   │   │   │   4
├─┼─┼─┼─┼─┼─┼─┼─┤
│   │   │   │ ○ │ ● │   │   │   │   5
├─┼─┼─┼─┼─┼─┼─┼─┤
│   │   │   │   │   │   │   │   │   6
├─┼─┼─┼─┼─┼─┼─┼─┤
│   │   │   │   │   │   │   │   │   7
├─┼─┼─┼─┼─┼─┼─┼─┤
│   │   │   │   │   │   │   │   │   8          ●have:1
└─┴─┴─┴─┴─┴─┴─┴─┘                  ○have:4
Please enter the coordinates for ●
```

3) Ask the player2 to enter coordinates for black3



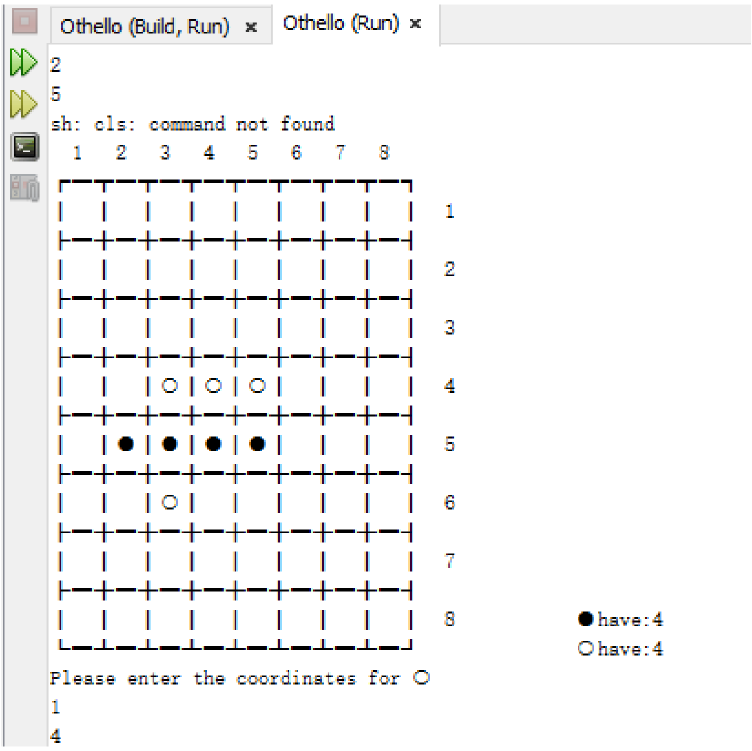
```
2
5
sh: cls: command not found
  1   2   3   4   5   6   7   8
┌─┬─┬─┬─┬─┬─┬─┬─┐
│ │ │ │ │ │ │ │ │  1
├─┼─┼─┼─┼─┼─┼─┼─┤
│ │ │ │ │ │ │ │ │  2
├─┼─┼─┼─┼─┼─┼─┼─┤
│ │ │ │ │ │ │ │ │  3
├─┼─┼─┼─┼─┼─┼─┼─┤
│ │ │○│○│○│ │ │ │  4
├─┼─┼─┼─┼─┼─┼─┼─┤
│ │●│●│●│●│ │ │ │  5
├─┼─┼─┼─┼─┼─┼─┼─┤
│ │ │○│ │ │ │ │ │  6
├─┼─┼─┼─┼─┼─┼─┼─┤
│ │ │ │ │ │ │ │ │  7
├─┼─┼─┼─┼─┼─┼─┼─┤
│ │ │ │ │ │ │ │ │  8          ●have:4
└─┴─┴─┴─┴─┴─┴─┴─┘             ○have:4
Please enter the coordinates for ○
1
4
```

4) When the game have someone win, output it.



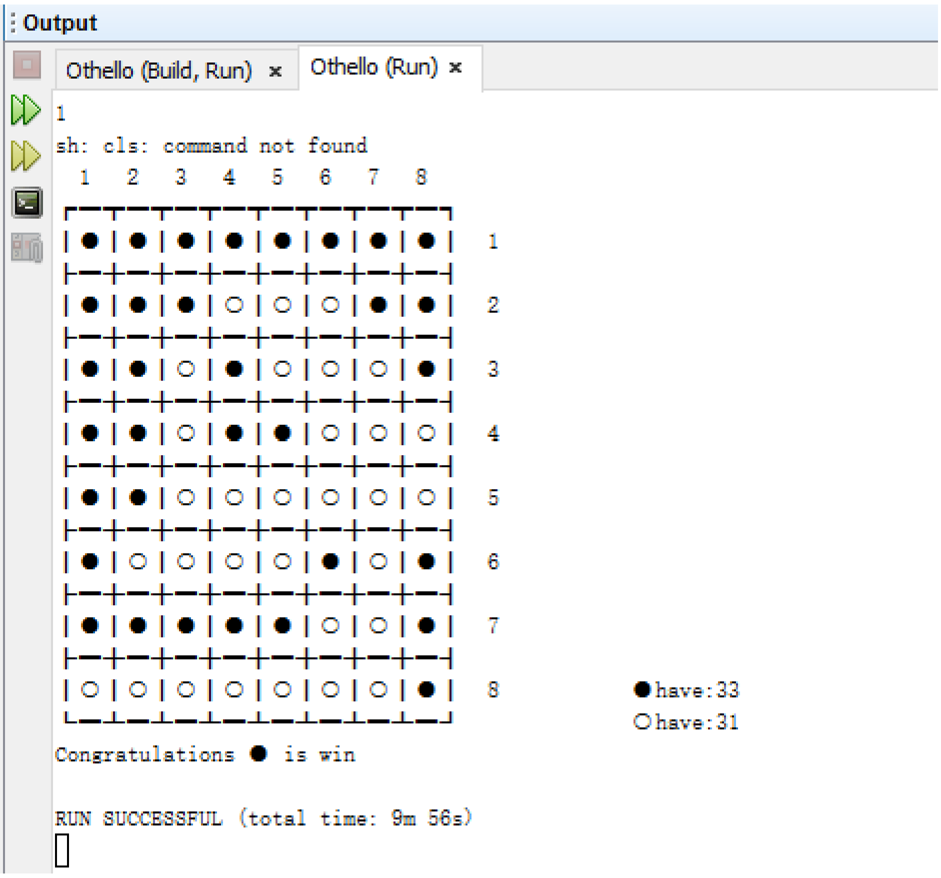
```
1
sh: cls: command not found
  1   2   3   4   5   6   7   8
┌─┬─┬─┬─┬─┬─┬─┬─┐
│●│●│●│●│●│●│●│●│  1
├─┼─┼─┼─┼─┼─┼─┼─┤
│●│●│●│○│○│○│●│●│  2
├─┼─┼─┼─┼─┼─┼─┼─┤
│●│●│○│●│○│○│○│●│  3
├─┼─┼─┼─┼─┼─┼─┼─┤
│●│●│○│●│●│○│○│○│  4
├─┼─┼─┼─┼─┼─┼─┼─┤
│●│●│○│○│○│○│○│○│  5
├─┼─┼─┼─┼─┼─┼─┼─┤
│●│○│○│○│○│●│○│●│  6
├─┼─┼─┼─┼─┼─┼─┼─┤
│●│●│●│●│●│○│○│●│  7
├─┼─┼─┼─┼─┼─┼─┼─┤
│○│○│○│○│○│○│○│●│  8          ●have:33
└─┴─┴─┴─┴─┴─┴─┴─┘             ○have:31
Congratulations ● is win

RUN SUCCESSFUL (total time: 9m 56s)
```

5) If it choices play AI, just want to enter the while



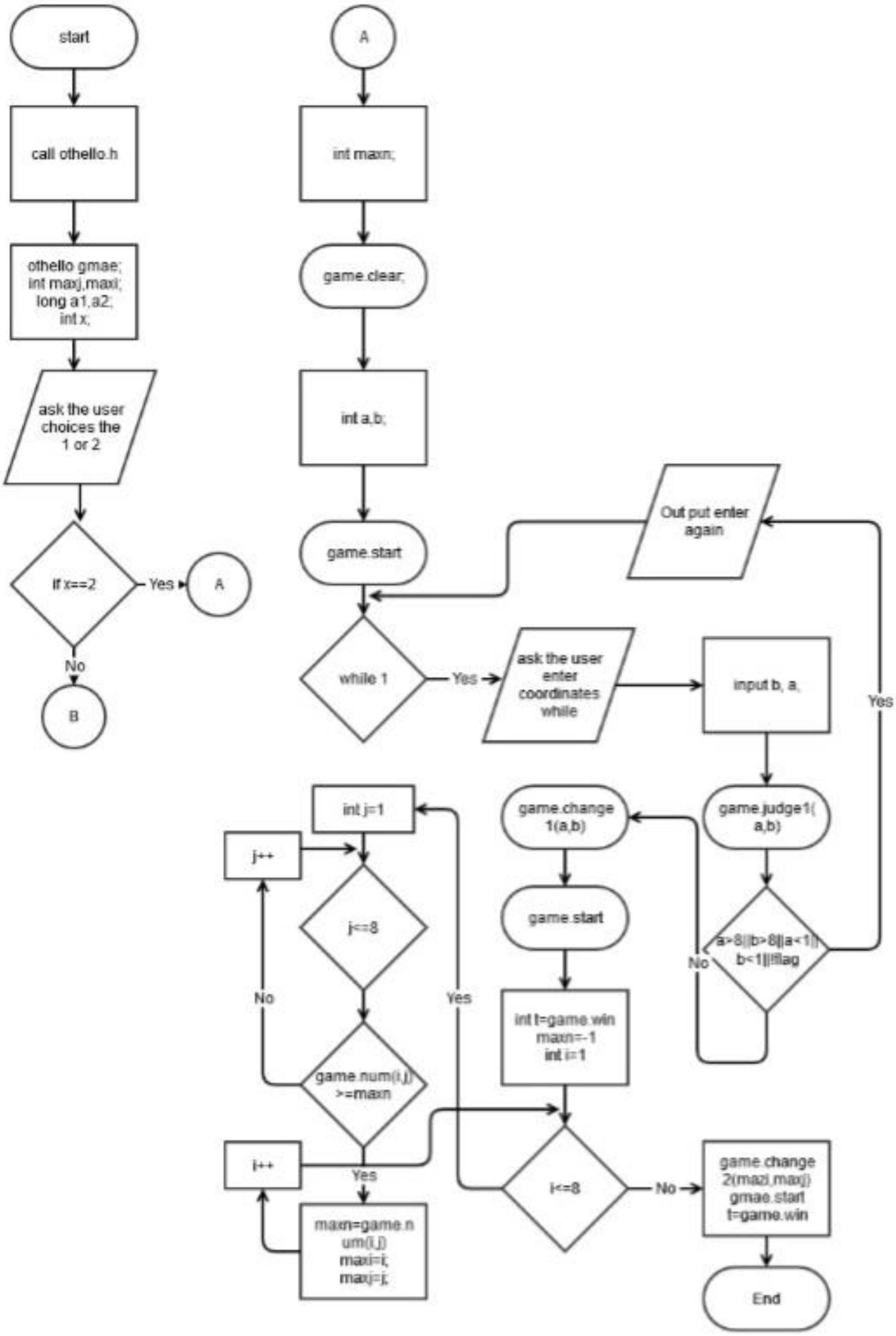
```
2
5
sh: cls: command not found
  1   2   3   4   5   6   7   8
┌───┬───┬───┬───┬───┬───┬───┬───┐
│   │   │   │   │   │   │   │   │   1
├───┼───┼───┼───┼───┼───┼───┼───┤
│   │   │   │   │   │   │   │   │   2
├───┼───┼───┼───┼───┼───┼───┼───┤
│   │   │   │   │   │   │   │   │   3
├───┼───┼───┼───┼───┼───┼───┼───┤
│   │   │ ○ │ ○ │ ○ │   │   │   │   4
├───┼───┼───┼───┼───┼───┼───┼───┤
│   │ ● │ ● │ ● │ ● │   │   │   │   5
├───┼───┼───┼───┼───┼───┼───┼───┤
│   │   │ ○ │   │   │   │   │   │   6
├───┼───┼───┼───┼───┼───┼───┼───┤
│   │   │   │   │   │   │   │   │   7
├───┼───┼───┼───┼───┼───┼───┼───┤
│   │   │   │   │   │   │   │   │   8          ●have:4
└───┴───┴───┴───┴───┴───┴───┴───┘              ○have:4
Please enter the coordinates for ○
1
4
```

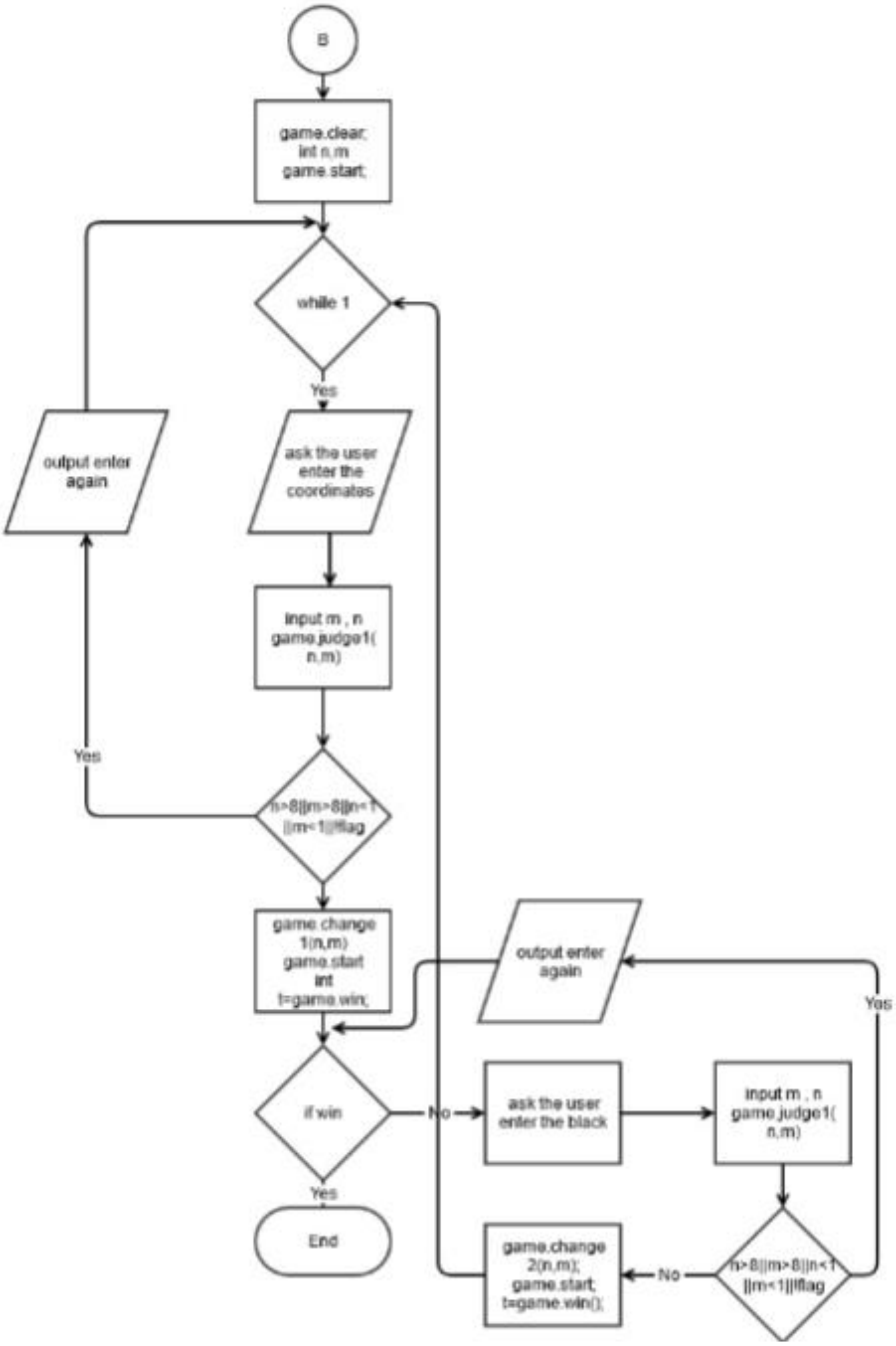
6) When have the winner for play with AI output it



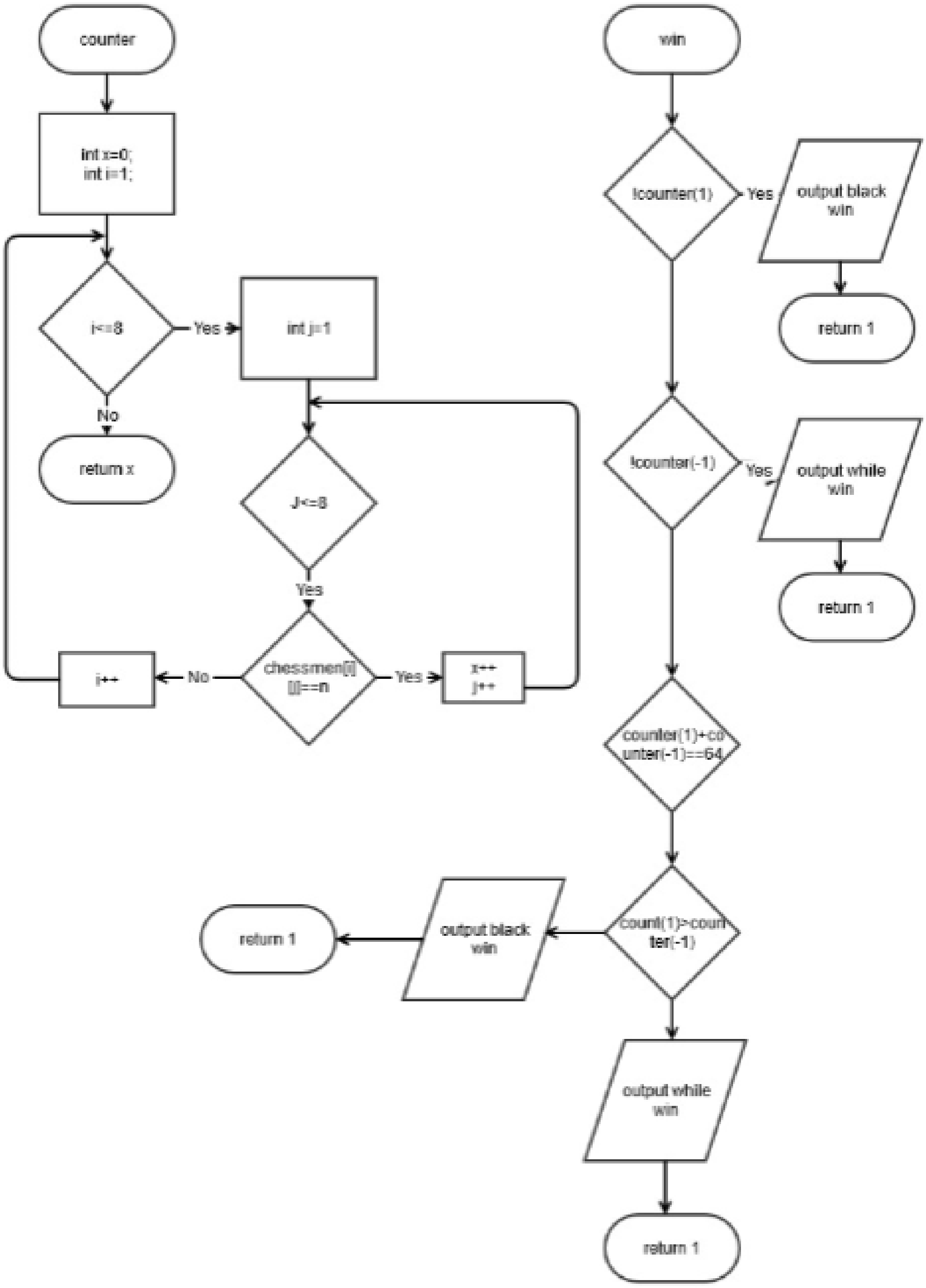
```
1
sh: cls: command not found
  1   2   3   4   5   6   7   8
┌───┬───┬───┬───┬───┬───┬───┬───┐
│ ● │ ● │ ● │ ● │ ● │ ● │ ● │ ● │   1
├───┼───┼───┼───┼───┼───┼───┼───┤
│ ● │ ● │ ● │ ○ │ ○ │ ○ │ ● │ ● │   2
├───┼───┼───┼───┼───┼───┼───┼───┤
│ ● │ ● │ ○ │ ● │ ○ │ ○ │ ○ │ ● │   3
├───┼───┼───┼───┼───┼───┼───┼───┤
│ ● │ ● │ ○ │ ● │ ● │ ○ │ ○ │ ○ │   4
├───┼───┼───┼───┼───┼───┼───┼───┤
│ ● │ ● │ ○ │ ○ │ ○ │ ○ │ ○ │ ○ │   5
├───┼───┼───┼───┼───┼───┼───┼───┤
│ ● │ ○ │ ○ │ ○ │ ○ │ ● │ ○ │ ● │   6
├───┼───┼───┼───┼───┼───┼───┼───┤
│ ● │ ● │ ● │ ● │ ● │ ○ │ ○ │ ● │   7
├───┼───┼───┼───┼───┼───┼───┼───┤
│ ○ │ ○ │ ○ │ ○ │ ○ │ ○ │ ○ │ ● │   8          ●have:33
└───┴───┴───┴───┴───┴───┴───┴───┘              ○have:31
Congratulations ● is win

RUN SUCCESSFUL (total time: 9m 56s)
```

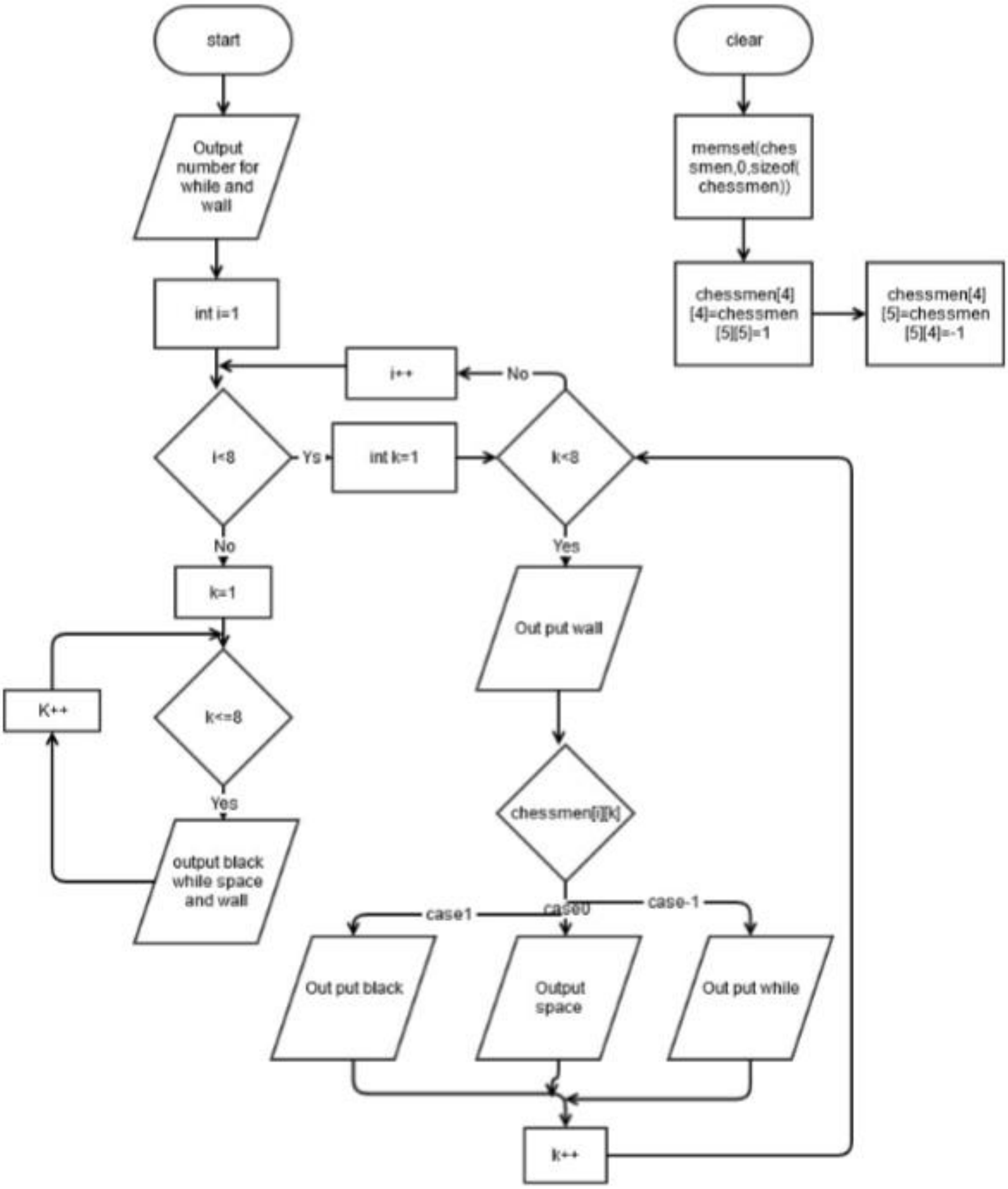
6. Flowchart(main)




B
game.clear;
int n,m
game.start;
while 1
Yes
ask the user enter the coordinates
output enter again
input m , n
game.judge1(n,m)
Yes
n>8||m>8||n<1 ||m<1||!flag
game.change 1(n,m)
game.start
int
t=game.win;
output enter again
Yes
if win
No
ask the user enter the black
input m , n
game.judge1(n,m)
Yes
End
game.change 2(n,m);
game.start;
t=game.win();
No
n>8||m>8||n<1 ||m<1||!flag

Flowchart (function)




start
Output number for while and wall
int i=1
i++
No
i<8
Ys
int k=1
k<8
No
k=1
K++
k<=8
Yes
output black while space and wall
Yes
Out put wall
chessmen[i][k]
case1
case0
case-1
Out put black
Output space
Out put while
k++
clear
memset(chessmen,0,sizeof(chessmen))
chessmen[4][4]=chessmen[5][5]=1
chessmen[4][5]=chessmen[5][4]=-1

7. Code

a) main

```
#include "Othello.h"
using namespace std;

int main(int argc, char** argv) {
        Othello game;
        int maxi,maxj;
        unsigned long    a1, a2;
        ios::sync_with_stdio(false);
        int x;

        //ask the user enter the number to chose play with person or AI
        cout<<"Welcome to the Othello,are you really to play the game?"<<endl;
        cout<<"Enter 1 to play the pvp game, enter 2 to play with AI!"<<endl;
        cin>>x;

        //if the user chose play with AI
        if(x==2)
        {
                int maxn;
                system("cls");        //clear
                //clear the checkerboard
                game.clear();
                int a, b;
                // out put the checkerboard
                game.start();
                while(1)
                {
                        //ask the user enter the coordinates
                        cout<<"Please enter the coordinates for  ○"<<endl;
                        expect1:
                        cin>>b>>a;
                        //call the function from the class to check the enter
                        int flag=game.judge1(a,b);

                        if(a>8||b>8||a<1||b<1||!flag)
                        {
                                cout<<"Unable to select, please enter again!"<<endl;
                                goto expect1;
                        }
                        else
                                //change the coordinate that the user
```

```
            //select to ○ from the checkerboard
            game.change1(a, b);
        system("cls");        //clear
        //out put the checkerboard
        game.start();
        //check the game have winner or not
        int t=game.win();
        if(t)
            break;
        //end it
        Sleep(2000);
        maxn=-1;
        for(int i=1;i<=8;i++)
        {
            for(int j=1;j<=8;j++)
            {
                if(game.num(i, j)>=maxn)
                {
                    maxn=game.num(i, j);
                    maxi=i;
                    maxj=j;
                }
            }
        }
        //change the coordinate that the AI
        //select to ● from the checkerboard
        game.change2(maxi,maxj);

        system("cls");        //clear
        //out put the checkerboard
        game.start();
        //check the game have winner or not
        t=game.win();
        if(t)
            break;

    }
}
else
{
    system("cls");        //clear
    //clear the checkerboard
    game.clear();
    int n, m;
```

```
// out put the checkerboard
game.start();
//when the user enter to play pvp game,
while(1)
{
    //ask the user enter the coordinates
    cout<<"Please enter the coordinates for  ○"<<endl;
    Electrics1:
    cin>>m>>n;
    //call the function from the class to check the enter
    int flag=game.judge1(n,m);
    if(n>8||m>8||n<1||m<1||!flag)
    {
        cout<<"Unable to select, please enter again!"<<endl;
        goto Electrics1;
    }
    else
        //change the coordinate that the user
        //select to  ○  from the checkerboard
        game.change1(n, m);
    system("cls");        //clear

    // out put the checkerboard
    game.start();
    //check the game have winner or not
    int t=game.win();

    //for the other player
    if(t)
        break;
    cout<<"Please enter the coordinates for  ●"<<endl;
    Electrics2:
    cin>>m>>n;
    //call the function from the class to check the enter
    flag=game.judge2(n,m);
    if(n>8||m>8||n<1||m<1||!flag)
    {
        cout<<"Unable to select, please enter again!"<<endl;
        goto Electrics2;
    }
    else
        //change the coordinate that the user
        //select to  ●  from the checkerboard
        game.change2(n, m);
```

```
                system("cls");        //clear
                // out put the checkerboard
                game.start();
                //check the game have winner or not
                t=game.win();
                if(t)
                    break;
            }
        }
        return 0;
}
```

b) Othello.h

```
    #ifndef OTHELLO_H
    #define OTHELLO_H

    //library
    #include<iostream>
    #include<algorithm>
    #include<windows.h>
    #include<cstring>
    #include<ctime>
    #include<cmath>
    #include<cstdio>
    using namespace std;


    class Othello{
    public:
        //the function to calculate the ● and ○ number
        int counter(int n);
        // out put the checkerboard
        void start();
        //check the game have winner or not
        int win();
        //clear the checkerboard
        void clear();
        //change the coordinate that the user
        //select to ○ from the checkerboard
        void change1(int x, int y);
        //change the coordinate that the user
        //select to ● from the checkerboard
        void change2(int x, int y);
        //check the enter for ○ select
        int judge1(int x, int y);
        //check the enter for ● select
        int judge2(int x, int y);
        //the function to calculate the AI select
        int num(int x, int y);
    private:
        // the array for the checkerboard
        int    chessmen[9][9];
    };

#endif /* OTHELLO_H */
```

c) Othello.cpp

```
#include "Othello.h"


//the function to calculate the ● and ○ number
int Othello::counter(int n)
{
    int x;
    x=0;
    for(int i=1;i<=8;i++)
    {
        for(int j=1;j<=8;j++)
        {
            if(chessmen[i][j]==n)
                x++;
        }
    }
    return x;
}


// out put the checkerboard
void Othello::start()
{
    //group the checkerboard
    cout<<"   1    2    3    4    5    6    7    8    "<<endl;
    cout<<" ┏━━┳━━┳━━┳━━┳━━┳━━┳━━┳━━┓ "<<endl;
    for(int i=1;i<8;i++)
    {
        for(int k=1;k<=8;k++)
        {
            cout<<" ┃ ";
            switch(chessmen[i][k])
            {
                case 0:cout<<"    ";break;
                case 1:cout<<"●";break;
                case -1:cout<<"○";break;
            }
        }
        cout<<" ┃    "<<i<<endl;
        cout<<" ┣━━";
        for(int j=1;j<8;j++)
        {
                cout<<"┼━━";
            }
            cout<<"┨ "<<endl;
        }
        for(int k=1;k<=8;k++)
        {
            cout<<" ┃ ";
            switch(chessmen[8][k])
            {
                case 0:cout<<" ";break;
                case 1:cout<<"●";break;
                case -1:cout<<"○";break;
            }
            if(chessmen[8][k]==0)
                cout<<" ";
        }
        //it also out put the number of the  ●  and  ○
        cout<<" ┃    8                    ●have:"<<counter(1)<<endl;    ;
        cout<<" ┗━━┻━━┻━━┻━━┻━━┻━━┻━━┻━━┛                                ○
"<<"have:"
                <<counter(-1)<<endl;
    }

    //check the game have winner or not
    int Othello::win()
    {
        if(!counter(1))
        {
            cout<<"Congratulations  ●  is win"<<endl;
            return 1;
        }
        else if(!counter(-1))
        {
            cout<<"Congratulations  ○  is win"<<endl;
            return 1;
        }
        else if(counter(1)+counter(-1)==64)
        {
            if(counter(1)>counter(-1))
            {
                cout<<"Congratulations  ●  is win"<<endl;
                return 1;
            }
            else
```

```
                {
                        cout<<"Congratulations ○ is win"<<endl;
                        return 1;
                }
        }
        return 0;
}


//clear the checkerboard
void Othello::clear()
{
        memset(chessmen,0,sizeof(chessmen));
        //set the ● is 1 and ○ is -1
        chessmen[4][4]=chessmen[5][5]=1;
        chessmen[4][5]=chessmen[5][4]=-1;
}


//change the coordinate that the user
//select to ○ from the checkerboard
void Othello::change1(int x, int y)
{
        chessmen[x][y]=-1;
        if(chessmen[x][y+1]==1)
        {
                for(int i=y+2;i<=8;i++)
                {
                        if(chessmen[x][i]==0)
                                break;
                        if(chessmen[x][i]==1)
                                continue;
                        if(chessmen[x][i]==-1)
                        {
                                for(int j=y+1;j<=i;j++)
                                        chessmen[x][j]=-1;
                                break;
                        }
                }
        }
        if(chessmen[x][y-1]==1)
        {
                for(int i=y-2;i>=1;i--)
                {
```

```
        if(chessmen[x][i]==0)
            break;
        if(chessmen[x][i]==1)
            continue;
        if(chessmen[x][i]==-1)
        {
            for(int j=y-1;j>=i;j--)
                chessmen[x][j]=-1;
            break;
        }
    }
}
if(chessmen[x-1][y]==1)
{
    for(int i=x-2;i>=1;i--)
    {
        if(chessmen[i][y]==0)
            break;
        if(chessmen[i][y]==1)
            continue;
        if(chessmen[i][y]==-1)
        {
            for(int j=x-1;j>=i;j--)
                chessmen[j][y]=-1;
            break;
        }
    }
}
if(chessmen[x+1][y]==1)
{
    for(int i=x+2;i<=8;i++)
    {
        if(chessmen[i][y]==0)
            break;
        if(chessmen[i][y]==1)
            continue;
        if(chessmen[i][y]==-1)
        {
            for(int j=x+1;j<=i;j++)
                chessmen[j][y]=-1;
            break;
        }
    }
}
```

```
if(chessmen[x+1][y+1]==1)
{
        int top=y+2;
        for(int i=x+2;i<=8;i++)
        {
                if(chessmen[i][top]==0)
                        break;
                if(chessmen[i][top]==1)
                        top++;
                if(chessmen[i][top]==-1)
                {
                        top=y+1;
                        for(int j=x+1;j<=i;j++)
                        {
                                chessmen[j][top]=-1;
                                top++;
                        }
                        break;
                }
        }
}
if(chessmen[x-1][y-1]==1)
{
        int top=y-2;
        for(int i=x-2;i>=1;i--)
        {
                if(chessmen[i][top]==0)
                        break;
                if(chessmen[i][top]==1)
                        top--;
                if(chessmen[i][top]==-1)
                {
                        top=y-1;
                        for(int j=x-1;j>=i;j--)
                        {
                                chessmen[j][top]=-1;
                                top--;
                        }
                        break;
                }
        }
}
if(chessmen[x-1][y+1]==1)
{
```

```
                int top=y+2;
                for(int i=x-2;i>=1;i--)
                {
                        if(chessmen[i][top]==0)
                                break;
                        if(chessmen[i][top]==1)
                                top++;
                        if(chessmen[i][top]==-1)
                        {
                                top=y+1;
                                for(int j=x-1;j>=i;j--)
                                {
                                        chessmen[j][top]=-1;
                                        top++;
                                }
                                break;
                        }
                }
        }
        if(chessmen[x+1][y-1]==1)
        {
                int top=y-2;
                for(int i=x+2;i<=8;i++)
                {
                        if(chessmen[i][top]==0)
                                break;
                        if(chessmen[i][top]==1)
                                top--;
                        if(chessmen[i][top]==-1)
                        {
                                top=y-1;
                                for(int j=x+1;j<=i;j++)
                                {
                                        chessmen[j][top]=-1;
                                        top--;
                                }
                                break;
                        }
                }
        }

}

//change the coordinate that the user
```

```
//select to ● from the checkerboard
void Othello::change2(int x, int y)
{
    chessmen[x][y]=1;
    if(chessmen[x][y+1]==-1)
    {
        for(int i=y+2;i<=8;i++)
        {
            if(chessmen[x][i]==0)
                break;
            if(chessmen[x][i]==-1)
                continue;
            if(chessmen[x][i]==1)
            {
                for(int j=y+1;j<=i;j++)
                    chessmen[x][j]=1;
                break;
            }
        }
    }
    if(chessmen[x][y-1]==-1)
    {
        for(int i=y-2;i>=1;i--)
        {
            if(chessmen[x][i]==0)
                break;
            if(chessmen[x][i]==-1)
                continue;
            if(chessmen[x][i]==1)
            {
                for(int j=y-1;j>=i;j--)
                    chessmen[x][j]=1;
                break;
            }
        }
    }
    if(chessmen[x-1][y]==-1)
    {
        for(int i=x-2;i>=1;i--)
        {
            if(chessmen[i][y]==0)
                break;
            if(chessmen[i][y]==-1)
                continue;
            if(chessmen[i][y]==1)
            {
                for(int j=x-1;j>=i;j--)
                    chessmen[j][y]=1;
                break;
            }
        }
    }
    if(chessmen[x+1][y]==-1)
    {
        for(int i=x+2;i<=8;i++)
        {
            if(chessmen[i][y]==0)
                break;
            if(chessmen[i][y]==-1)
                continue;
            if(chessmen[i][y]==1)
            {
                for(int j=x+1;j<=i;j++)
                    chessmen[j][y]=1;
                break;
            }
        }
    }
    if(chessmen[x+1][y+1]==-1)
    {
        int top=y+2;
        for(int i=x+2;i<=8;i++)
        {
            if(chessmen[i][top]==0)
                break;
            if(chessmen[i][top]==-1)
                top++;
            if(chessmen[i][top]==1)
            {
                top=y+1;
                for(int j=x+1;j<=i;j++)
                {
                    chessmen[j][top]=1;
                    top++;
                }
                break;
            }
        }
```

```
}
if(chessmen[x-1][y-1]==-1)
{
        int top=y-2;
        for(int i=x-2;i>=1;i--)
        {
                if(chessmen[i][top]==0)
                        break;
                if(chessmen[i][top]==-1)
                        top--;
                if(chessmen[i][top]==1)
                {
                        top=y-1;
                        for(int j=x-1;j>=i;j--)
                        {
                                chessmen[j][top]=1;
                                top--;
                        }
                        break;
                }
        }
}
if(chessmen[x-1][y+1]==-1)
{
        int top=y+2;
        for(int i=x-2;i>=1;i--)
        {
                if(chessmen[i][top]==0)
                        break;
                if(chessmen[i][top]==-1)
                        top++;
                if(chessmen[i][top]==1)
                {
                        top=y+1;
                        for(int j=x-1;j>=i;j--)
                        {
                                chessmen[j][top]=1;
                                top++;
                        }
                        break;
                }
        }
}
if(chessmen[x+1][y-1]==-1)
```

```
		{
			int top=y-2;
			for(int i=x+2;i<=8;i++)
			{
				if(chessmen[i][top]==0)
					break;
				if(chessmen[i][top]==-1)
					top--;
				if(chessmen[i][top]==1)
				{
					top=y-1;
					for(int j=x+1;j<=i;j++)
					{
						chessmen[j][top]=1;
						top--;
					}
					break;
				}
			}
		}

	}


	//check the enter for ○ select
	int Othello::judge1(int x, int y)
	{
		if(chessmen[x][y]==1||chessmen[x][y]==-1)
			return 0;
		//the ○ back side is ●
		if(chessmen[x][y+1]==1)
		{
			for(int i=y+2;i<=8;i++)
			{
				if(chessmen[x][i]==0)
					break;
				if(chessmen[x][i]==1)
					continue;
				if(chessmen[x][i]==-1)
					return 1;
			}
		}
		//the ● front side is ○
		if(chessmen[x][y-1]==1)
```

```
{
        for(int i=y-2;i>=1;i--)
        {
                if(chessmen[x][i]==0)
                        break;
                if(chessmen[x][i]==1)
                        continue;
                if(chessmen[x][i]==-1)
                        return 1;
        }
}
//the ● top side is ○
if(chessmen[x-1][y]==1)
{
        for(int i=x-2;i>=1;i--)
        {
                if(chessmen[i][y]==0)
                        break;
                if(chessmen[i][y]==1)
                        continue;
                if(chessmen[i][y]==-1)
                        return 1;
        }
}
//the ● lower side is ○
if(chessmen[x+1][y]==1)
{
        for(int i=x+2;i<=8;i++)
        {
                if(chessmen[i][y]==0)
                        break;
                if(chessmen[i][y]==1)
                        continue;
                if(chessmen[i][y]==-1)
                        return 1;
        }
}
//the ● right down side is ○
if(chessmen[x+1][y+1])
{
        int top=y+2;
        for(int i=x+2;i<=8;i++)
        {
                if(chessmen[i][top]==0)
```

```
            break;
        if(chessmen[i][top]==1)
            top++;
        if(chessmen[i][top]==-1)
            return 1;

    }
}
//the ● left up side is ○
if(chessmen[x-1][y-1])
{
    int top=y-2;
    for(int i=x-2;i>=1;i--)
    {
        if(chessmen[i][top]==0)
            break;
        if(chessmen[i][top]==1)
            top--;
        if(chessmen[i][top]==-1)
            return 1;
    }
}
//the ● right up side is ○
if(chessmen[x-1][y+1])
{
    int top=y+2;
    for(int i=x-2;i>=1;i--)
    {
        if(chessmen[i][top]==0)
            break;
        if(chessmen[i][top]==1)
            top++;
        if(chessmen[i][top]==-1)
            return 1;
    }
}
//the ● left down side is ○
if(chessmen[x+1][y-1])
{
    int top=y-2;
    for(int i=x+2;i<=8;i++)
    {
        if(chessmen[i][top]==0)
            break;
            if(chessmen[i][top]==1)
                top--;
            if(chessmen[i][top]==-1)
                return 1;
        }
    }
    return 0;

}


//check the enter for ● select
int Othello::judge2(int x, int y)
{
    if(chessmen[x][y]==1||chessmen[x][y]==-1)
        return 0;
    //the ● back side is ○
    if(chessmen[x][y+1]==-1)
    {
        for(int i=y+2;i<=8;i++)
        {
            if(chessmen[x][i]==0)
                break;
            if(chessmen[x][i]==-1)
                continue;
            if(chessmen[x][i]==1)
                return 1;
        }
    }
    //the ● front side is ○
    if(chessmen[x][y-1]==-1)
    {
        for(int i=y-2;i>=1;i--)
        {
            if(chessmen[x][i]==0)
                break;
            if(chessmen[x][i]==-1)
                continue;
            if(chessmen[x][i]==1)
                return 1;
        }
    }
    //the ● top side is ○
    if(chessmen[x-1][y]==-1)
```

```
{
        for(int i=x-2;i>=1;i--)
        {
                if(chessmen[i][y]==0)
                        break;
                if(chessmen[i][y]==-1)
                        continue;
                if(chessmen[i][y]==1)
                        return 1;
        }
}
//the ● lower side is ○
if(chessmen[x+1][y]==-1)
{
        for(int i=x+2;i<=8;i++)
        {
                if(chessmen[i][y]==0)
                        break;
                if(chessmen[i][y]==-1)
                        continue;
                if(chessmen[i][y]==1)
                        return 1;
        }
}
//the ● right down side is ○
if(chessmen[x+1][y+1]==-1)
{
        int top=y+2;
        for(int i=x+2;i<=8;i++)
        {
                if(chessmen[i][top]==0)
                        break;
                if(chessmen[i][top]==-1)
                        top++;
                if(chessmen[i][top]==1)
                        return 1;
        }
}
//the ● left up side is ○
if(chessmen[x-1][y-1]==-1)
{
        int top=y-2;
        for(int i=x-2;i>=1;i--)
        {
            if(chessmen[i][top]==0)
                break;
            else if(chessmen[i][top]==-1)
                top--;
            if(chessmen[i][top]==1)
                return 1;
        }
    }
    //the ● right up side is ○
    if(chessmen[x-1][y+1]==-1)
    {
        int top=y+2;
        for(int i=x-2;i>=1;i--)
        {
            if(chessmen[i][top]==0)
                break;
            if(chessmen[i][top]==-1)
                top++;
            if(chessmen[i][top]==1)
                return 1;
        }
    }
    //the ● left down side is ○
    if(chessmen[x+1][y-1]==-1)
    {
        int top=y-2;
        for(int i=x+2;i<=8;i++)
        {
            if(chessmen[i][top]==0)
                break;
            if(chessmen[i][top]==-1)
                top--;
            if(chessmen[i][top]==1)
                return 1;
        }
    }
    return 0;
}


//the function to calculate the AI select
int Othello::num(int x, int y)
{
    if(!judge2(x, y))
```

```
        return 0;
int n=0;
chessmen[x][y]=1;
//the ● back side is ○
if(chessmen[x][y+1]==-1)
{
        for(int i=y+2;i<=8;i++)
        {
                if(chessmen[x][i]==0)
                        break;
                if(chessmen[x][i]==-1)
                        continue;
                if(chessmen[x][i]==1)
                {
                        for(int j=y+1;j<i;j++)
                                n++;
                        break;
                }
        }
}
//the ● front side is ○
if(chessmen[x][y-1]==-1)
{
        for(int i=y-2;i>=1;i--)
        {
                if(chessmen[x][i]==0)
                        break;
                if(chessmen[x][i]==-1)
                        continue;
                if(chessmen[x][i]==1)
                {
                        for(int j=y-1;j>i;j--)
                                n++;
                        break;
                }
        }
}
//the ● top side is ○
if(chessmen[x-1][y]==-1)
{
        for(int i=x-2;i>=1;i--)
        {
                if(chessmen[i][y]==0)
                        break;
        if(chessmen[i][y]==-1)
            continue;
        if(chessmen[i][y]==1)
        {
            for(int j=x-1;j>i;j--)
                n++;
            break;
        }
    }
}
//the ● lower side is ○
if(chessmen[x+1][y]==-1)
{
    for(int i=x+2;i<=8;i++)
    {
        if(chessmen[i][y]==0)
            break;
        if(chessmen[i][y]==-1)
            continue;
        if(chessmen[i][y]==1)
        {
            for(int j=x+1;j<i;j++)
                n++;
            break;
        }
    }
}
//the ● right down side is ○
if(chessmen[x+1][y+1]==-1)
{
    int top=y+2;
    for(int i=x+2;i<=8;i++)
    {
        if(chessmen[i][top]==0)
            break;
        if(chessmen[i][top]==-1)
            top++;
        if(chessmen[i][top]==1)
        {
            top=y+1;
            for(int j=x+1;j<i;j++)
            {
                n++;
                top++;
            }
            break;
        }
    }
}
//the ● left up side is ○
if(chessmen[x-1][y-1]==-1)
{
    int top=y-2;
    for(int i=x-2;i>=1;i--)
    {
        if(chessmen[i][top]==0)
            break;
        if(chessmen[i][top]==-1)
            top--;
        if(chessmen[i][top]==1)
        {
            top=y-1;
            for(int j=x-1;j>i;j--)
            {
                n++;
                top--;
            }
            break;
        }
    }
}
//the ● right up side is ○
if(chessmen[x-1][y+1]==-1)
{
    int top=y+2;
    for(int i=x-2;i>=1;i--)
    {
        if(chessmen[i][top]==0)
            break;
        if(chessmen[i][top]==-1)
            top++;
        if(chessmen[i][top]==1)
        {
            top=y+1;
            for(int j=x-1;j>i;j--)
            {
                n++;
                top++;
                }
                break;
            }
        }
    }
    //the ● left down side is ○
    if(chessmen[x+1][y-1]==-1)
    {
        int top=y-2;
        for(int i=x+2;i<=8;i++)
        {
            if(chessmen[i][top]==0)
                break;
            if(chessmen[i][top]==-1)
                top--;
            if(chessmen[i][top]==1)
            {
                top=y-1;
                for(int j=x+1;j<i;j++)
                {
                    n++;
                    top--;
                }
                break;
            }
        }
    }
    chessmen[x][y]=0;
    return n;
}
```