Five in Row Game

Yanwan Ye

CSC5

FALL 2016

48102

Introduction

Five in row game is a game which is played between two persons. The rule is the players put chess pieces one by one. For example, player 1 represents number 1 and he/she moves a piece in the chess board. Then player 2 who represents number 2 is turned to play. If one of them puts five pieces in the same line, he/ she will be the winner. The code is trying to block your competitor’s road at the right time.

How the Five in Row Game Works

Object of the Game

Chess board

Rules of the Game

1. Show a blank chess board which is 16 * 16.
2. Player 1 moves in a piece.
3. Player 2 moves in another piece. If this lace has a piece already, he/she should re-enter the value.

4. If the player enters a value is not between 1 ~ 16, he/she still need re-enter the value.
5. Show the chess board again until one of them win the game.
6. Ask if the players want to play the game again.

My Approach to the Game

While thinking about how I can program this game, I got couple questions:

1. How to show my chess board?
2. How can I put my array into the chess board?
3. How can I know five pieces is in the line?
4. How can I determine who the winner is?

The Logic of it All

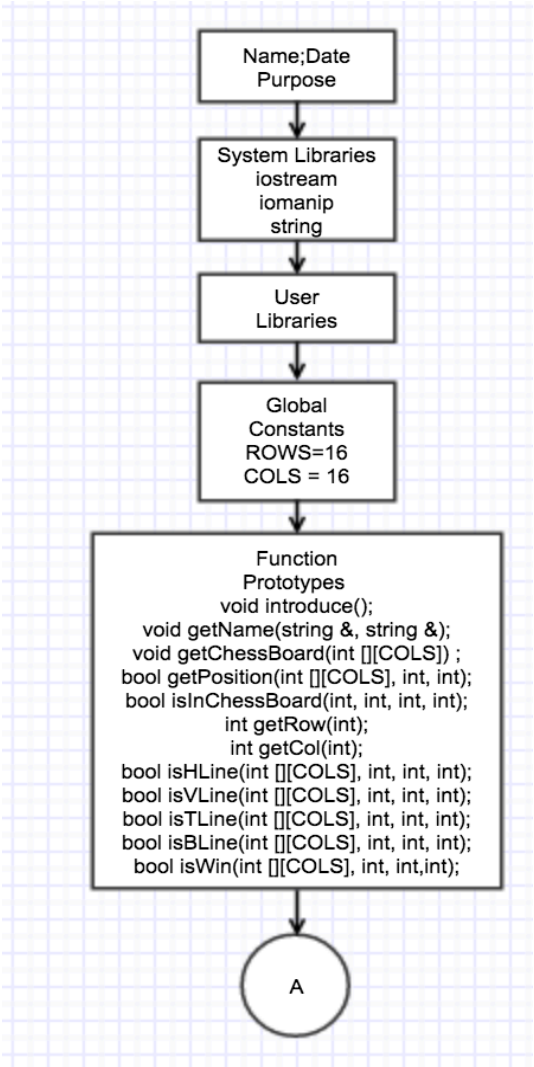
Rules of the Game

Since my flowchart is extremely long, I will break it up into smaller pieces and accompany it with pseudocode here. To view my complete flowchart, please visit:

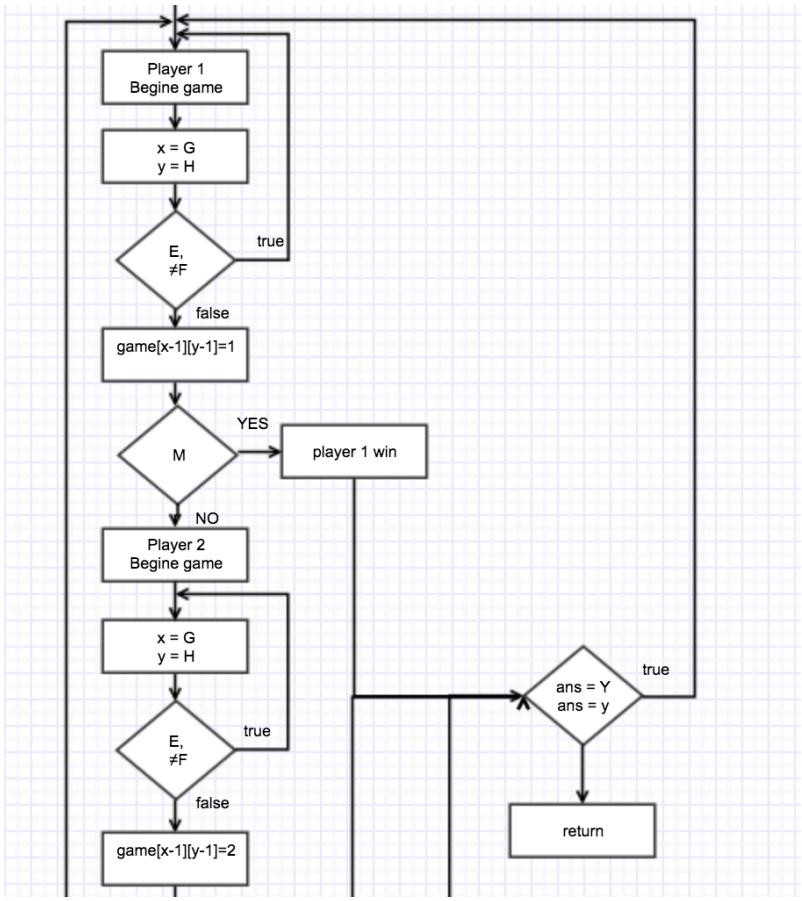
https://www.gliffy.com/go/publish/11299071

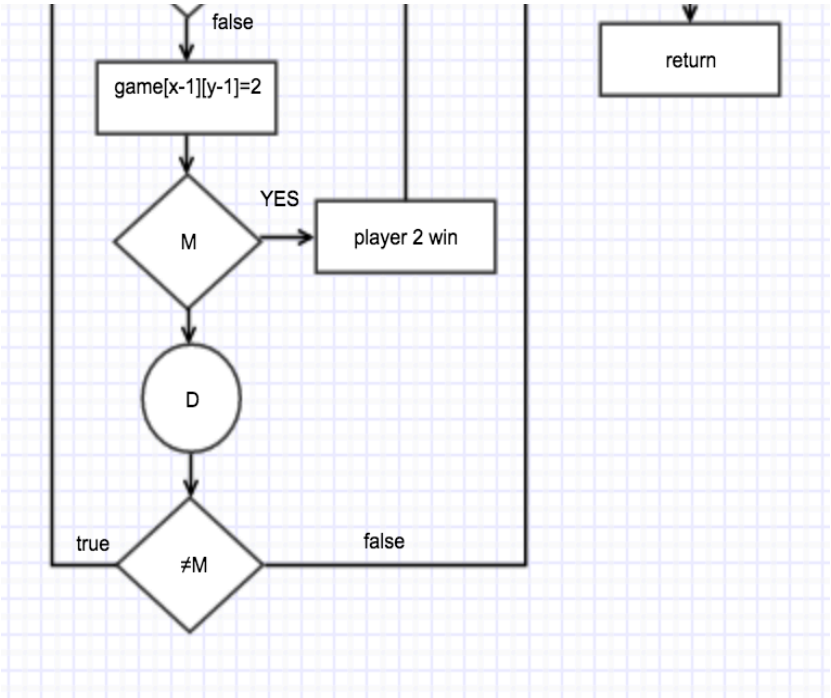
System Libraries declare: iostream, iomanip, string and fstream.

Declare global constants and all prototypes functions.

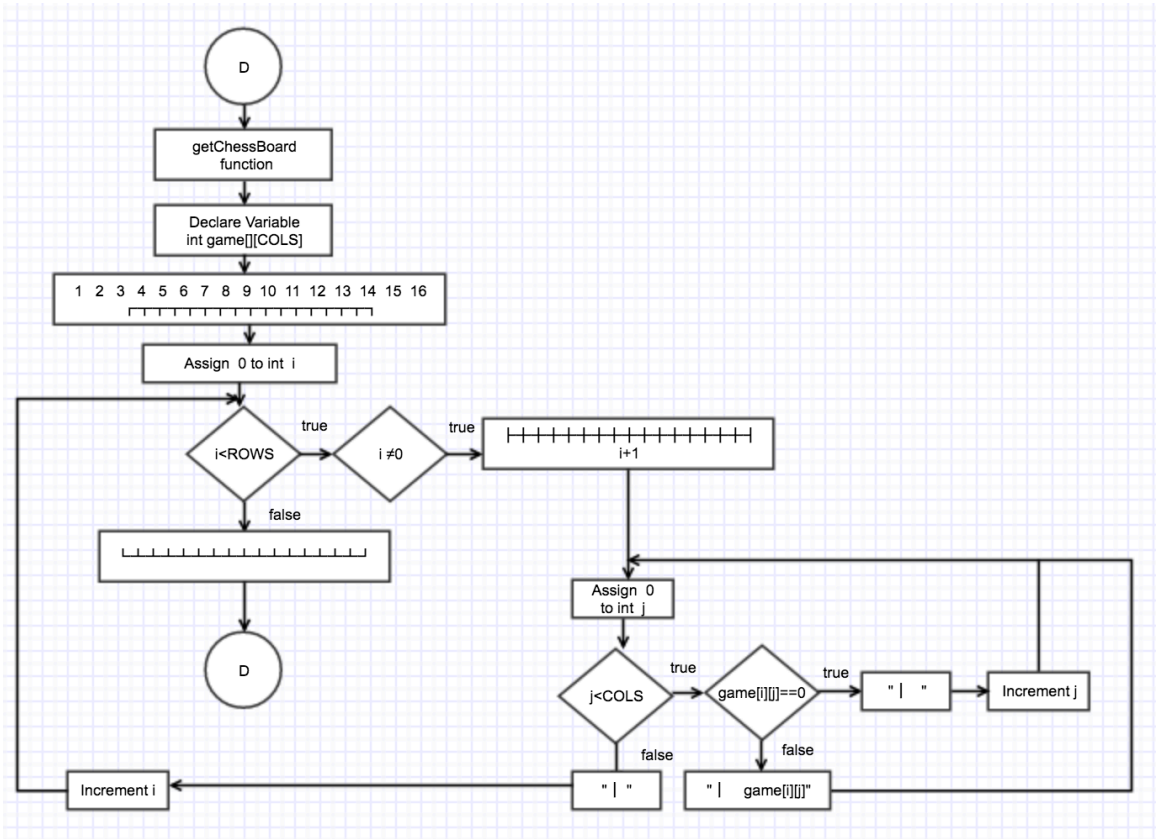


The whole program to play the game:



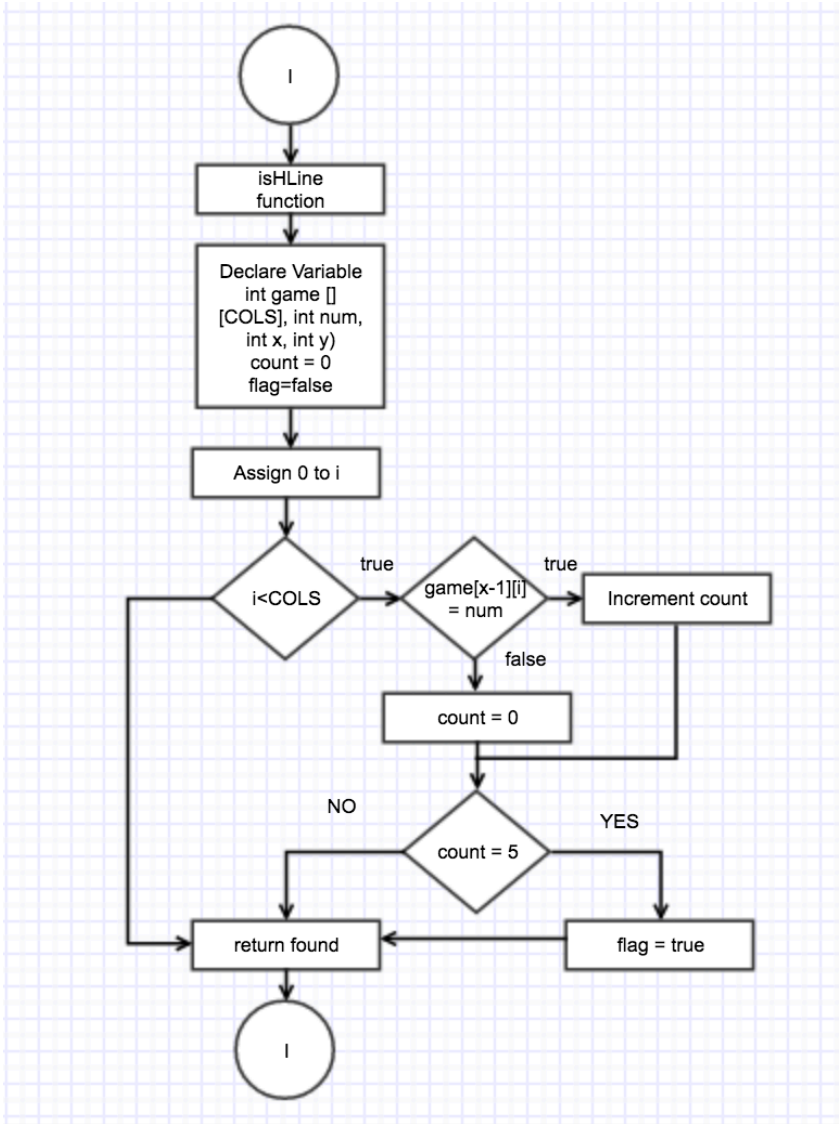


Chess board:


D
getChessBoard function
Declare Variable int game[][COLS]
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Assign 0 to int i
i<ROWS
true
i ≠0
true
i+1
false
D
Assign 0 to int j
j<COLS
true
game[i][j]==0
true
" | "
Increment j
false
false
" | "
" | game[i][j]"
Increment i

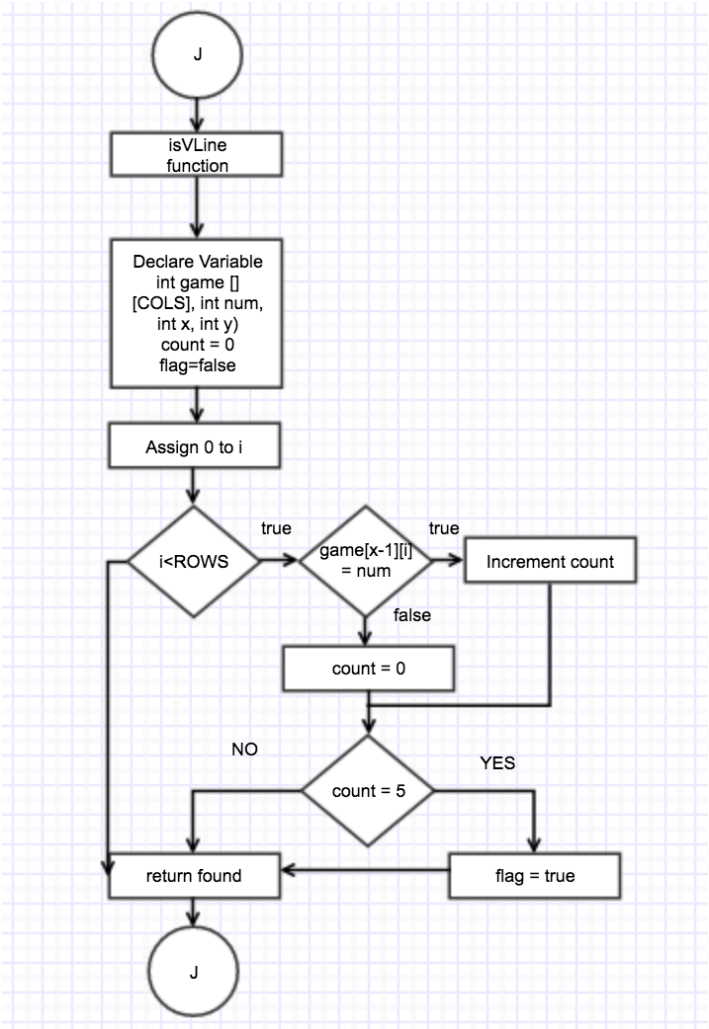
Horizon:

Determines whether this place have a piece is five in the horizon or not.



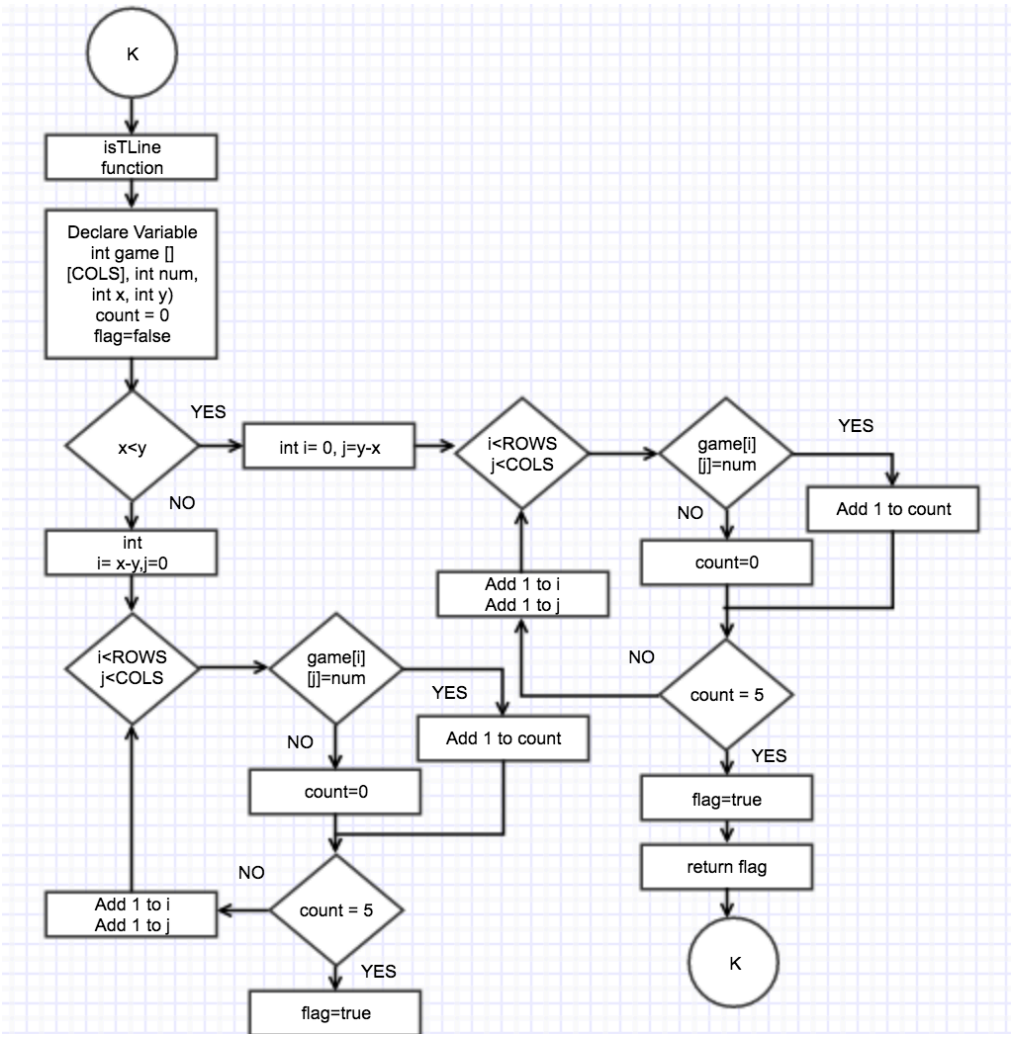
Vertical:

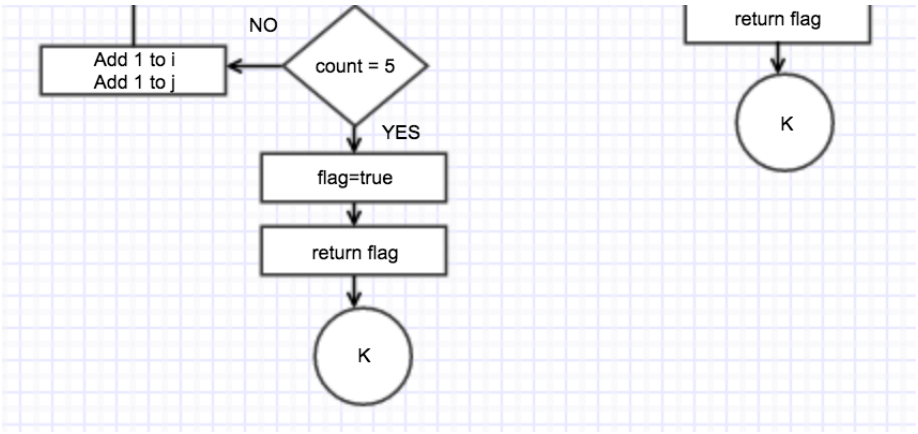
Determines whether this place have a piece is five in the vertical or not



Left top to right bottom:

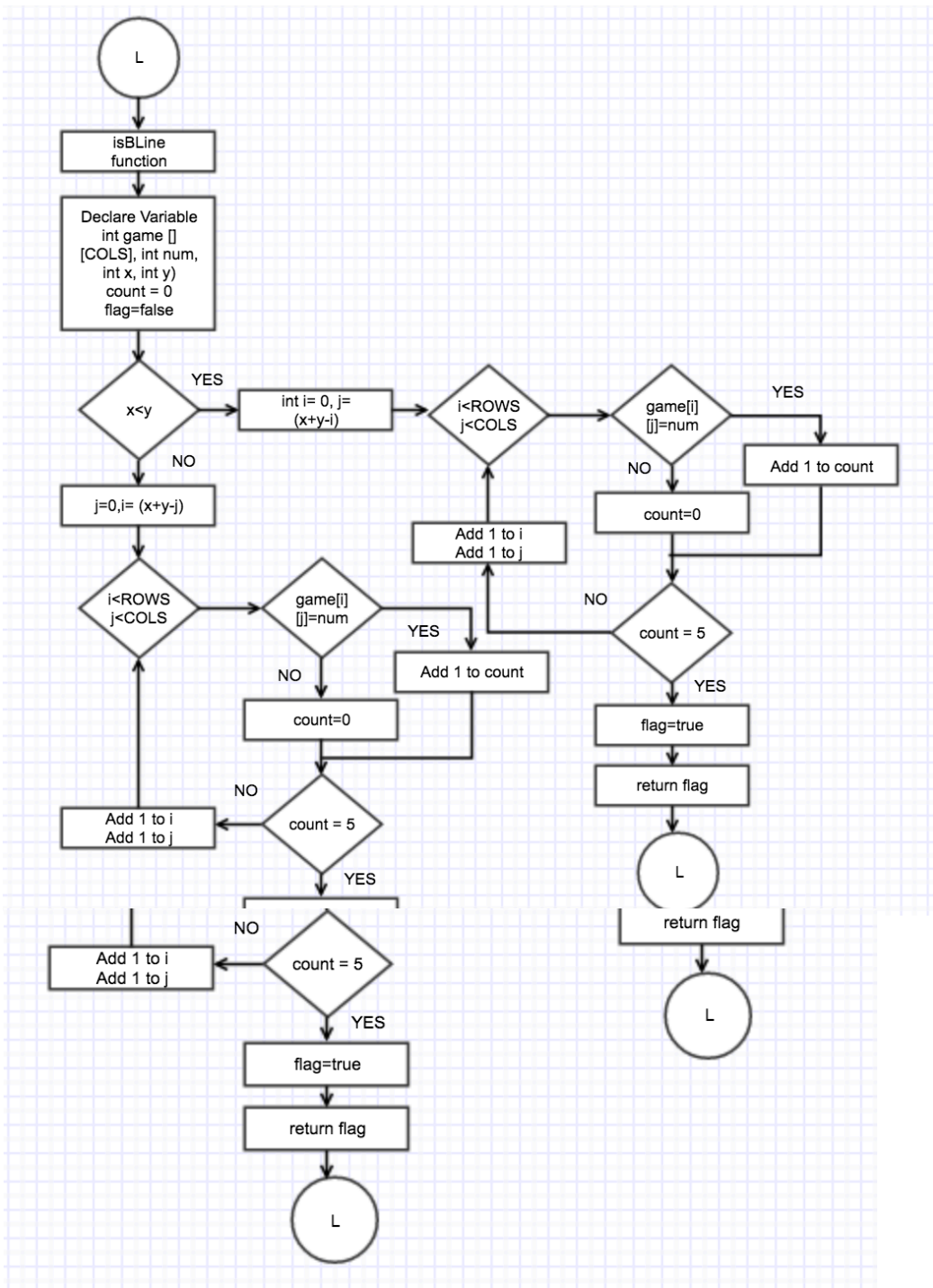
Determines whether this place have a piece is five from left top to right bottom or not



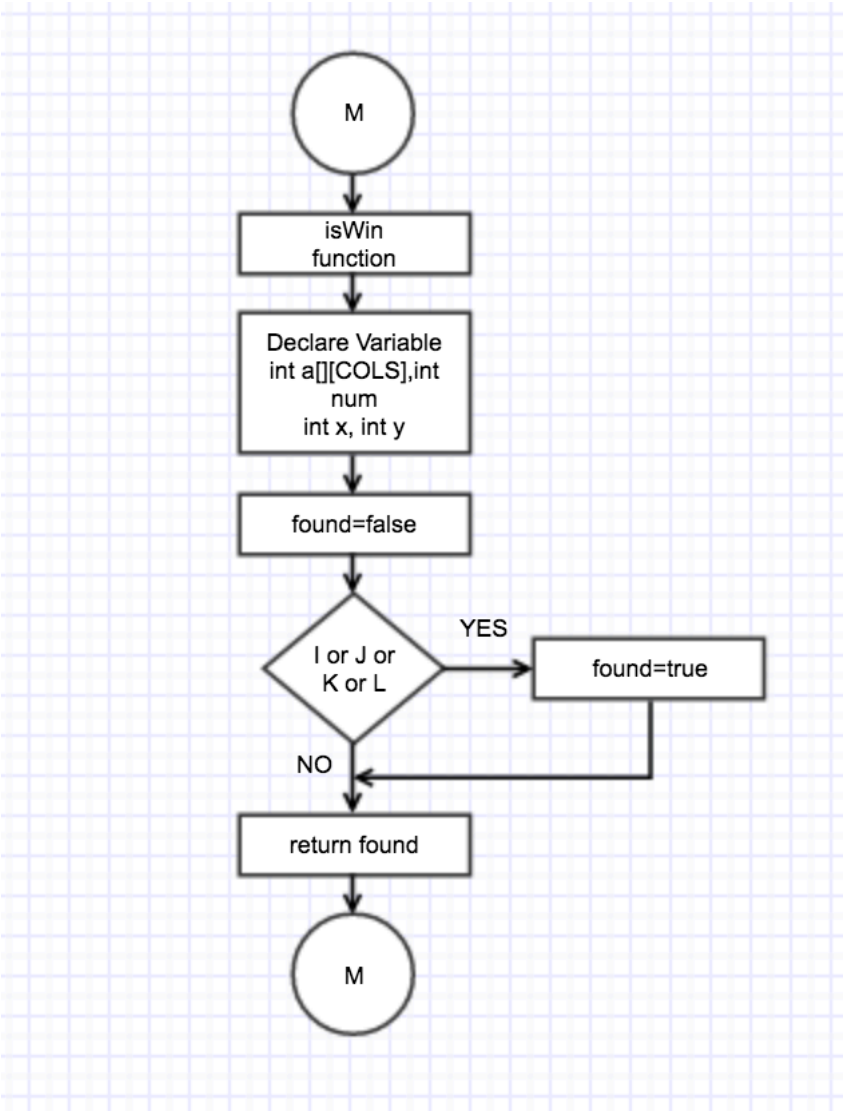


Left bottom to right top:

Determines whether this place have a piece is five from left bottom to right top or not



Determine if someone win the game



Constructs & Concepts Utilized

iostream Library

Name	Frequency	Description	Location
const int	2	ROWS, COLS	Line15,16
cout	26	Output Data	Throughout
cin	5	Input Data	Throughout
&	2	Reference	Line129,136
bool	16	Determine value is true or false	Throughout

iomanip Library

Name	Frequency	Description	Location
setw(2)	1	Format the row number	Line162

string Library

Name	Frequency	Description	Location
string	2	Declare players name	Line129,136

fstream Library

Name	Frequency	Description	Location
ftream	6	Write the win player to the file	48,49,76,79,103,106

Data Types:

Data Types	Frequency	Location
int	4	Throughout
char	1	Line98
string	2	Line129,136

Conditional Statements:

Conditional Statement	Frequency	Starting Location
if	13	Line 65,84,158,190,207,261,286,314,331,362,373,379,400
if/else	7	Line 167,256,280,305,308,353,356

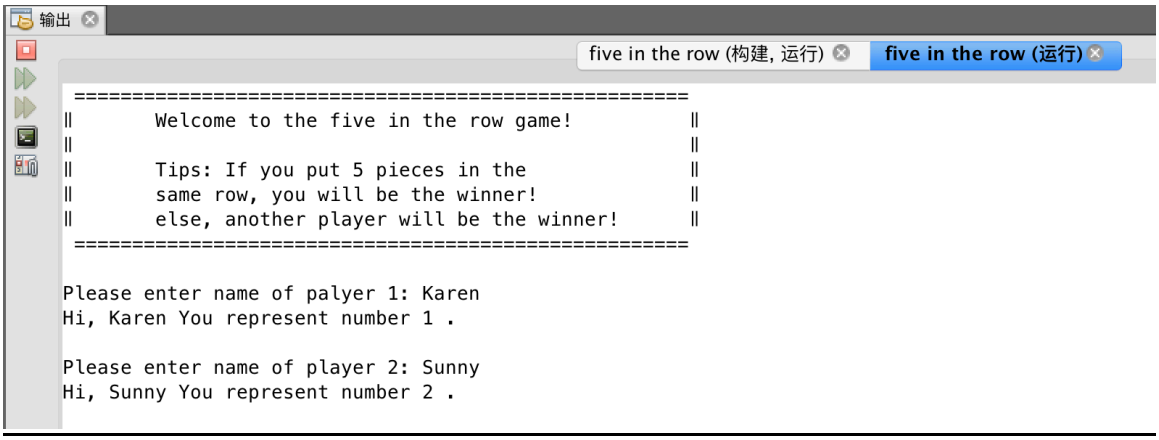
Loops:

Loops	Frequency	Starting Location
for	13	Line 155,165,255,279
while	7	Line 307,324,355,372
do-while	2	Line 48,51,56,74

Function Prototype:

Function Prototype	Starting Location
void introduce()	Line 109
void getName(string &, string &)	Line 125
void getChessBoard(int [][COLS])	Line 148
bool getPosition(int [][COLS], int, int)	Line 186
bool isInChessBoard(int, int, int, int)	Line 203
int getRow(int)	Line 220
int getCol(int)	Line 234
bool isHLine(int [][COLS], int, int, int)	Line 250
bool isVLine(int [][COLS], int, int, int)	Line 275
bool isTLine(int [][COLS], int, int, int)	Line 300
bool isBLine(int [][COLS], int, int, int)	Line 348
bool isWin(int [][COLS], int, int,int)	Line 395

Proof of a Working Product



输出

five in the row (运行)　five in the row (构建, 运行

```
Hi, Sunny You represent number 2 .

    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Karen move in a piece:
enter the row:5
enter the column:7
```

输出

five in the row (运行)　five in the row (构建, 运

```
enter the row:9
enter the column:11

    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16
1
2       2   2   2                   2   2
3
4
5                               1
6                               1
7                               1   1
8                               1
9                                   1
10
11
12
13
14
15
16

Congratulation! Karen win.

Do you want to play again(Y/N):
```

Program:

```
/*
 * File:   main.cpp
 * Author: yeyanwan
 *
 * Created on December 8, 2016, 5:43 PM
 * Purpose:  This program is used to play a five in row game.
 */

#include <iostream>
#include <string>
#include <iomanip>
#include <fstream>

using namespace std;

// Constant for the array size
const int ROWS=16;
const int COLS=16;

// Function prototype
void introduce();
void getName(string &, string &);
void getChessBoard(int [][COLS]) ;
bool getPosition(int [][COLS], int, int);
bool isInChessBoard(int, int, int, int);
int getRow(int);
int getCol(int);
bool isHLine(int [][COLS], int, int, int);
bool isVLine(int [][COLS], int, int, int);
bool isTLine(int [][COLS], int, int, int);
bool isBLine(int [][COLS], int, int, int);
bool isWin(int [][COLS], int, int,int);

/*
 *
 */
```

起始页 main.cpp

源 历史记录

```cpp
/*
 *
 */
int main(int argc, char** argv) {

    int game[ROWS][COLS]={};                //Array of chess board
    int x, y;                               //The player put in a row and column
    int row,col;                            //Row and column
    string player1,player2;                 //Name of player1 and player2
    char ans;                               //Answer of play again

    introduce();                            //Introduce Function
    getName(player1,player2);               //Get the players' names Function
    getChessBoard(game);                    //Get the chess board Function

    ofstream outputFile;                   // File stream object
    outputFile.open("win.txt");            // Open a file named win.txt.


    //Play the game again
    do{
        int game[ROWS][COLS]={};
        //Player1 and player2 turn to put in pieces
        do{
            //Player 1 begine to play
            cout<<"\n"<<player1<<" move in a piece: "<<endl;

            //Check the place
            do{
                x=getRow(row);
                y=getCol(col);
            }while(getPosition(game, x,y) || !(isInChessBoard(ROWS, COLS, x, y)));

            //Set the place
            game[x-1][y-1]=1;

            //Get the winner is Player1
            if(isWin(game, 1, x, y)){
                getChessBoard(game);
```

```
            getChessBoard(game);

            cout<<"\nCongratulation! "<<player1<<" win.\n";

            // Write the win player to the file.
            outputFile<<"\nCongratulation! "<<player1<<" win.\n";

            //Close the file.
            outputFile.close();

            break;
        }

        //Player 1 begine to play
        cout<<"\n"<<player2<<" move in a piece: "<<endl;

        //Check the place
        do{
            x=getRow(row);
            y=getCol(col);

        }while(getPosition(game, x,y) || !(isInChessBoard(ROWS, COLS, x, y)));

        //Set the place
        game[x-1][y-1]=2;

        //Get the winner is Player2
        if(isWin(game, 2, x, y)){
            getChessBoard(game);
            cout<<"\nCongratulation! "<<player2<<" win.\n";

            // Write the win player to the file.
            outputFile<<"\nCongratulation! "<<player2<<" win.\n";

            //Close the file.
            outputFile.close();

            break;
            }
```

源 历史记录

```
                    break;
                    }

                //Display the chess board
                getChessBoard(game);

            }while((!isWin(game, 1, x, y)) && (!isWin(game, 2, x, y)));

            //Does the player want to play the game again
            cout<<"\nDo you want to play again(Y/N): ";
            cin>>ans;

        }while(ans == 'Y' || ans == 'y');

        return 0;
    }

    // ************************************************************
    // The introduce function is used to welcome the players    *
    // and introduces this game.                                *
    // This function display a message.                         *
    // ************************************************************

    void introduce()
    {
        cout<<" ====================================================\n";
        cout<<"‖       Welcome to the five in the row game!         ‖\n";
        cout<<"‖                                                    ‖\n";
        cout<<"‖       Tips: If you put 5 pieces in the             ‖\n";
        cout<<"‖       same row, you will be the winner!            ‖\n";
        cout<<"‖       else, another player will be the winner!     ‖\n";
        cout<<" ====================================================\n";
    }

    // ************************************************************
    // The getName function is used to get the players' names.  *
    // This function display a message.                         *
    // ************************************************************
```

源 历史记录

```
147
148 void getName(string &player1, string &player2)
149 {
150     //Get play1 's name
151     cout<<"\nPlease enter name of palyer 1: ";
152     cin>>player1;
153
154     //Let the player be looked as number 1
155     cout<<"Hi, "<<player1<<" You represent number 1 .\n\n";
156
157     //Get paly2's name
158     cout<<"Please enter name of player 2: ";
159     cin>>player2;
160
161     //Let the player be looked as number 2
162     cout<<"Hi, "<<player2<<" You represent number 2 .\n";
163
164 }
165
166 // ************************************************************
167 // The getChessBoard function display the chess board       *
168 // and put valid array inside.                              *
169 // ************************************************************
170
171 void getChessBoard(int game[][COLS])
172 {
173     //Display the heading
174     cout<<"\n    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16\n";
175     cout<<"  ┌───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┬───┐"<<endl;
176
177     //Set the rows
178     for(int i=0; i<ROWS; i++)
179     {
180         //Skip the first row
181         if(i !=0)
182             cout<<"  ├───┼───┼───┼───┼───┼───┼───┼───┼───┼───┼───┼───┼───┼───┼───┼───┤"<<endl;
183
184             //Display the row number
185             cout<<left<<setw(2)<<i+1;
186
```

```
            cout<<left<<setw(2)<<i+1;

        //Set the column
        for(int j=0; j<COLS; j++){

            if(game[i][j]==0){
                cout<<"|   ";
            }
            else
                cout<<"| "<<game[i][j]<<" ";
        }
        cout<<"|"<<endl;
    }
    //Set the bottom
    cout<<"  └───┴───┴───┴
```

源 历史记录

```
225
226 bool isInChessBoard(int row, int col, int x, int y){
227
228     bool found=false;                                   //Flag to indiciate if the value was found
229
230     if(x>0 && x<(row+1) && y>0 && y<(col+1))            //Make enter the row and column is between 1~16
231         found=true;                                     //Set the flag
232
233     return found;                                       //Return the found
234 }
235
236 // ************************************************************
237 // The getRow function has an argument row as the which row *
238 // the player put in a piece. It returns row.               *
239 // ************************************************************
240
241 int getRow(int row){
242
243     //Get the row
244     cout<<"enter the row:";
245     cin>>row;
246
247     return row;
248 }
249
250 // ************************************************************
251 // The getCol function has an argument col as the which     *
252 // column the player put in a piece. It returns col.        *
253 // ************************************************************
254
255 int getCol(int col){
256
257     //Get the column
258     cout<<"enter the column:";
259     cin>>col;
260
261     return col;
262 }
```

源 历史记录

```cpp

// ************************************************************
// The isHLine function determines whether this place have  *
// a piece is five in the horizon or not.It has game array, *
// num of 1 or 2, row and column.                           *
// It is a bool fuction and returns true.                   *
// ************************************************************

bool isHLine(int game [][COLS], int num, int x, int y){
    int count = 0;                                  //Set a count to 0
    bool flag=false;                                //Flag to indiciate if the value was found

    //Determine the five in horizon
    for(int i=0;i<COLS;i++){
        if(game[x-1][i] == num){
            count++;
        }else{
            count = 0;
        }
        if(count==5){
            flag=true;                              //Set the flag
        }
    }
    return flag;                                   //Return the flag
}

// ************************************************************
// The isVLine function determines whether this place have  *
// a piece is five in the vertical or not.It has game       *
// array, num of 1 or 2, row and column.                    *
// It is a bool fuction and returns true.                   *
// ************************************************************

bool isVLine(int game [][COLS], int num, int x, int y){
    int count = 0;                                 //Set a count to 0
    bool flag=false;                               //Flag to indiciate if the value was found

```

源 历史记录

```
    bool flag=false;                          //Flag to indiciate if the value was found

    for(int i=0;i<ROWS;i++){
        if(game[i][y-1] == num){
            count++;
        }
        else{
            count = 0;
        }
        if(count==5){
            flag=true;                        //Set the flag
        }
    }
    return flag;                              //Return the flag
}

// ************************************************************
// The isTLine function determines whether this place have  *
// a piece is five from left top to right bottom or not.    *
// It has game array, num of 1 or 2, row and column.        *
// It is a bool fuction and returns true.                   *
// ************************************************************

bool isTLine(int game [][COLS], int num, int x, int y){

    int count = 0;                            //Set a count to 0
    bool flag=false;                          //Flag to indiciate if the value was found

    if(x<y){
        int i= 0,j=y-x;
        while(i<(ROWS+1) && j<(COLS+1)){
            if(game[i][j]==num){
                count++;
            }
            else{
                count = 0;
            }
            if(count == 5){
                flag=true;                    //Set the flag
            }
```

起始页 main.cpp

源 历史记录

```cpp
                flag=true;                      //Set the flag
            }
            i++;
            j++;
        }
        return flag;                            //Return the flag
    }
    else{
        int i= x-y,j=0;
        while(i<(ROWS+1) && j<(COLS+1)){
            if(game[i][j]==num){
                count++;
            }
            else{
                count = 0;
            }
            if(count == 5){
                flag=true;                      //Set the flag
            }
            i++;
            j++;
        }
        return flag;                            //Return the flag
    }
}

// ************************************************************
// The isBLine function determines whether this place have  *
// a piece is five from left bottom to right top or not.    *
// It has game array, num of 1 or 2, row and column.        *
// It is a bool fuction and returns true.                   *
// ************************************************************

bool isBLine(int game [][COLS], int num, int x, int y){

    int count = 0;                          //Set a count to 0
    bool flag=false;                        //Flag to indiciate if the value was found
```

源 历史记录

```
bool isBLine(int game [][COLS], int num, int x, int y){

    int count = 0;                          //Set a count to 0
    bool flag=false;                        //Flag to indiciate if the value was found

    if(x<y){
        int i= 1,j=((x+y)-i);
        while(i<(ROWS+1) && j<(COLS+1)){
            if(game[i][j]==num){
                count++;
            }
            else{
                count = 0;
            }
            if(count == 5){
                flag=true;                //Set the flag
            }
            i++;
        }
        return flag;                     //Return the flag
    }
    else{
        int j=1,i= ((x+y)-j);
        while(i<(ROWS+1) && j<(COLS+1)){
            if(game[i][j]==num){
                count++;
            }
            else{
                count = 0;
            }
            if(count == 5){
                flag=true;              //Set the flag
            }
            j++;
        }
        return flag;                   //Return the flag
    }
}

```

源 历史记录

```
                if(count == 5){
                    flag=true;              //Set the flag
                }
                j++;
            }
            return flag;                    //Return the flag
        }
    }

    // ************************************************************
    // The isWin function determines whether one of the         *
    // players is win.                                          *
    // It is a bool fuction and returns true.                   *
    // ************************************************************

    bool isWin(int game[][COLS], int num, int x,int y){

        bool found=false;                   //Flag to indiciate if the value was found

        //Determine if one play is win when one line is in five
        if (isHLine(game,num, x, y) || isVLine(game,num, x, y)
            ||isTLine(game,num, x, y) || isBLine(game,num, x, y) ){

            found=true;                     //Set the flag
        }
        return found;                       //Return the flag
    }
```