Project 2

<Small Sudoku>

CSC 5 - 48102

Professor: Mark Lehr

Student: Weikang Du

Project Content Menu

1. Introduction

The small sudoku game is a logical number game, the player need to fill out all the blank of 3X3 chart by using 1-9, each number is only able to use once. Player need to finish this chart and make the sum of number in each row and each column equal 15 to get the basic winning. The perfect solution is that the sum of each slant equal 15 and based on basic winning.

In the game process, firstly, there will display a 3X3 chart with a random number between 1-9 in random place, then the player is able to fill one number each time, and also the player can clear one position each time after one move.

Furthermore, this game is a version of alpha, therefore, we have 10 registered ID for tester. Player need to enter in the valid ID to start the game. And the result will export to a record file.

2. Code Summary

Total line of code :	573
Comment line :	about 117
Number of variables:	15
Number of function:	13

Back Menu

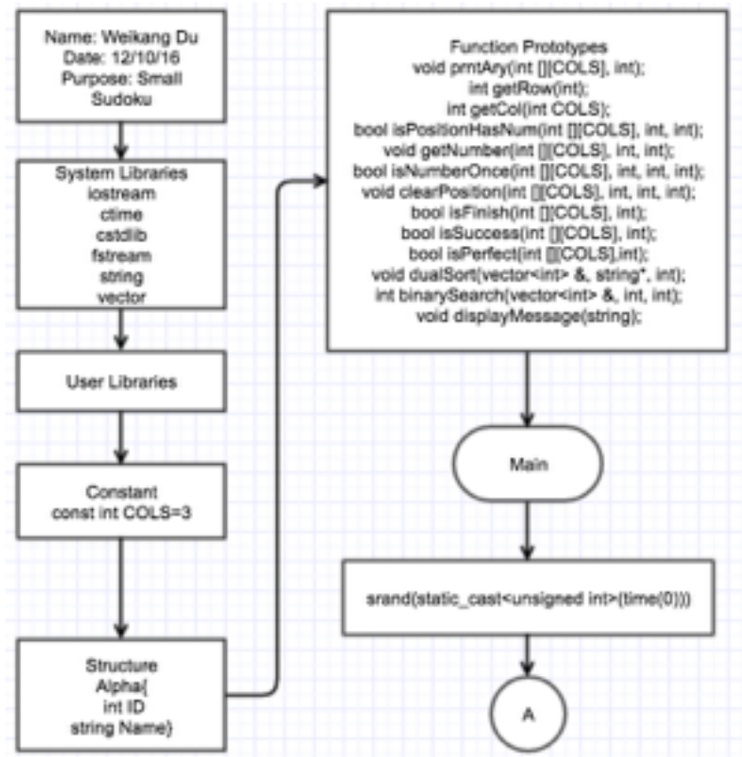
Constructs	
Do while loop	line 96, 109, 263
While loop	line 82, 87, 119, 129, 143, 202, 408, 415, 472, 516, 541
If	line 160, 163, 268, 295, 320, 321, 356, 379, 382, 448, 452, 494, 564, 568
If else	line 63, 151, 152, 170, 265, 566
Switch Case	line 184, 195
For loop	line 67, 159, 161, 290, 294, 345, 350, 377, 378, 446, 447, 563, 565

File and random	
Srand	line 44
Rand	line 103, 104, 107
Ifstream	line 51
Ofstream	line 50
Open( )	line 60, 61
Close( )	line 72, 211

Back Menu

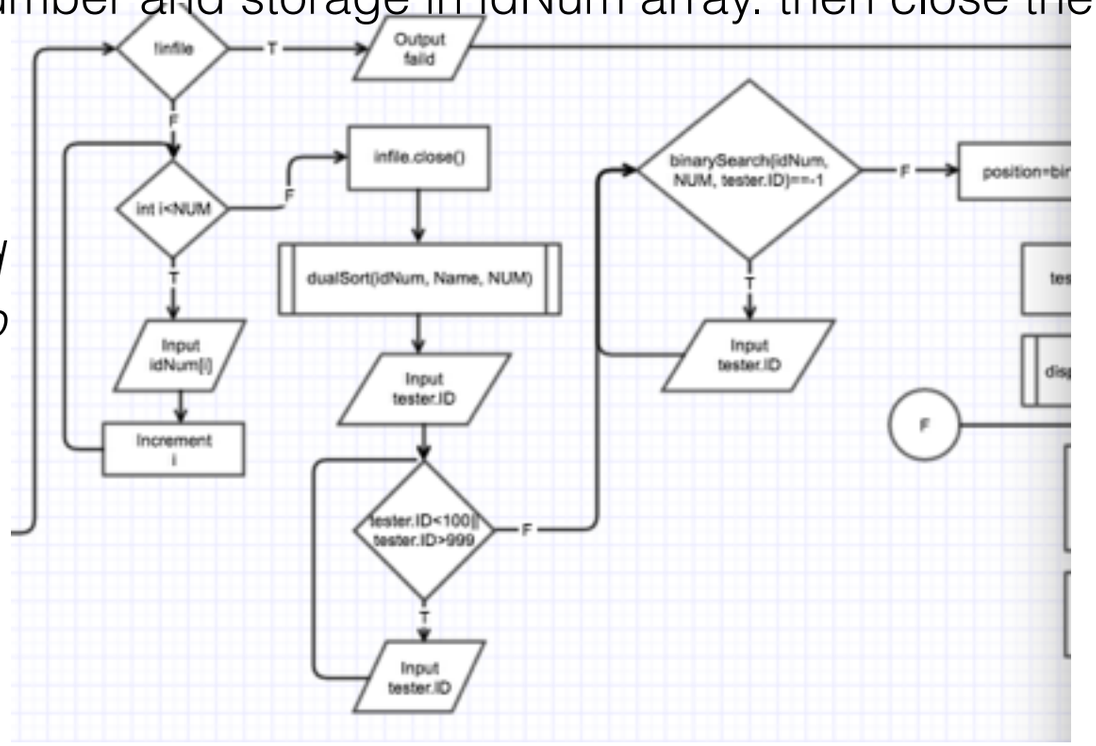
3. Psudo Code

Opening comment and declare the system libraries. Set the constant int number for column size of the array. Declare a structure for tester's information. And then function prototypes. And main function begin, then set the random seed immediately.



At here, open the file, and using for loop, to get in all id number and storage in idNum array. then close the file

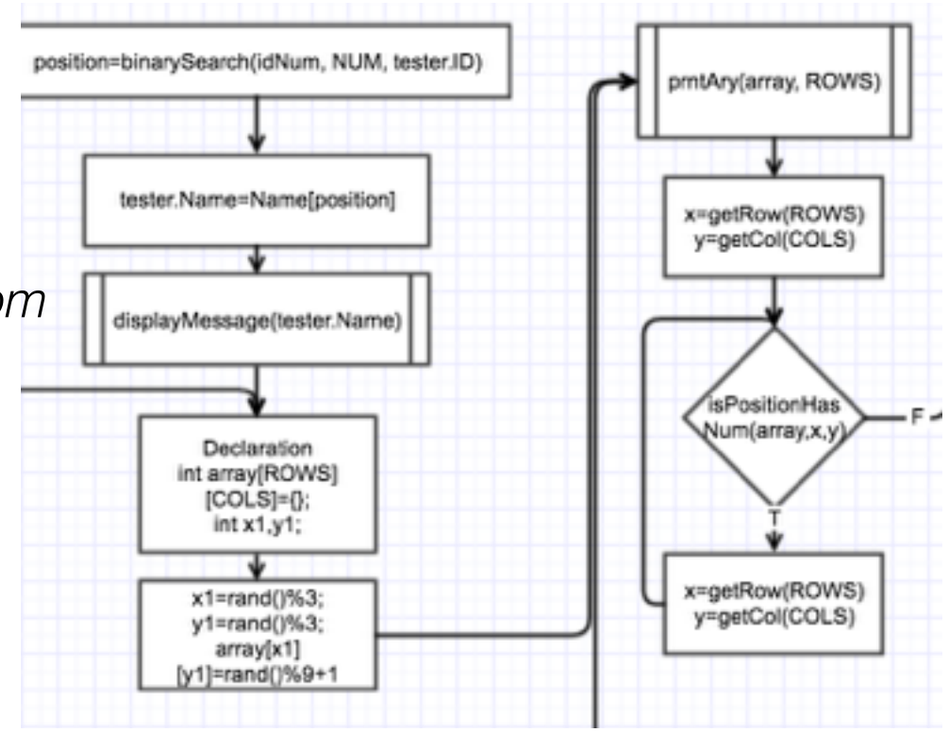
Declare all variables, then open the file for output and read. Verify opening or not. Let the player enter in the id, then do the dual sorting for integer number id array and parallel name array. Then valid the id and do the binary search to get the position of this id.



Back Menu

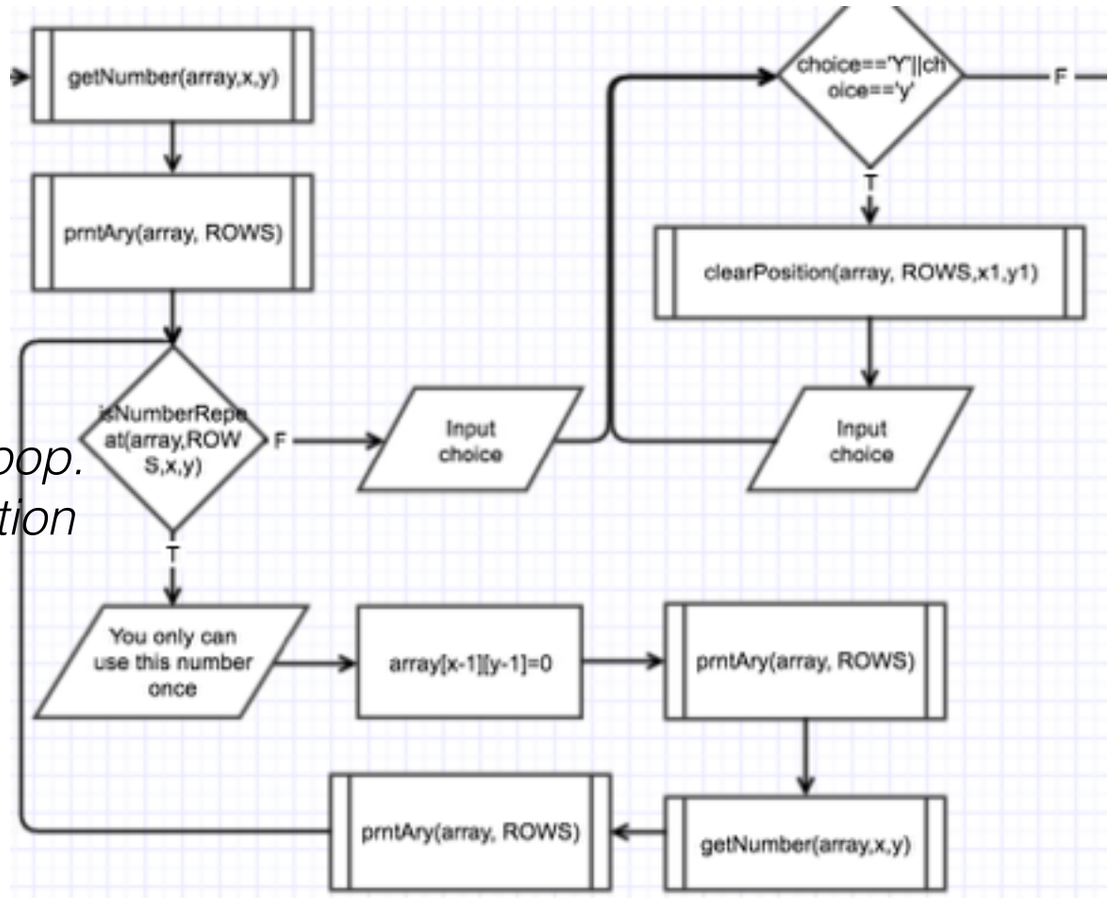
Plug the position in the name array, and pass it to displayMessage function.
Set the answer array is null. Random a position with a random number between 1-9.
Then display the output. Using the function
get the position which the user want to put number in.

Valid the position, if there already has a number, player can not enter in a number in this position.



If not, then input a number and display the output valid the number repeat or not. if it dose, then clear it and display, then input another number and display in a loop.

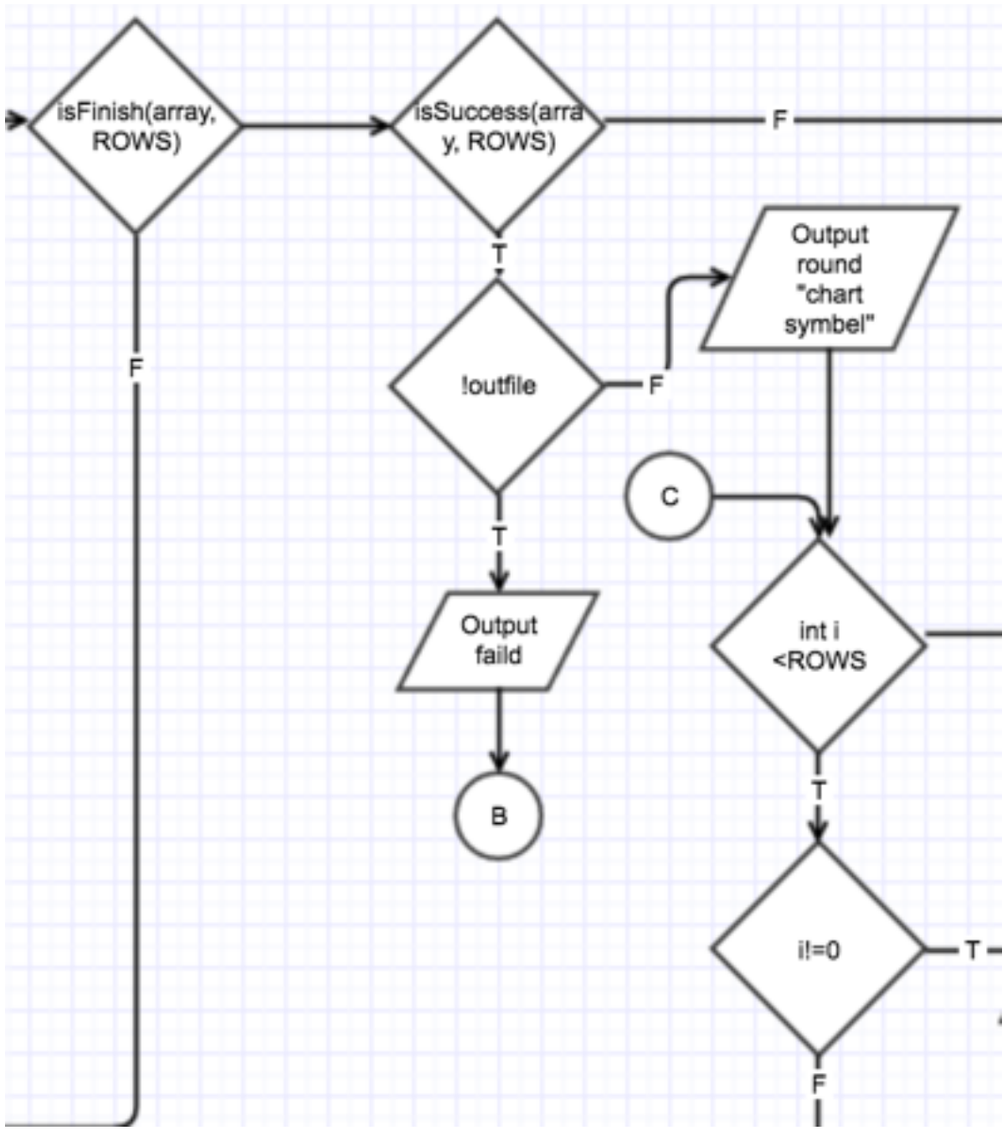
Until the number is once, user can get a chance to decide to clear one position each time or not in a while loop. if the player want to clear, then call the clearPosition function

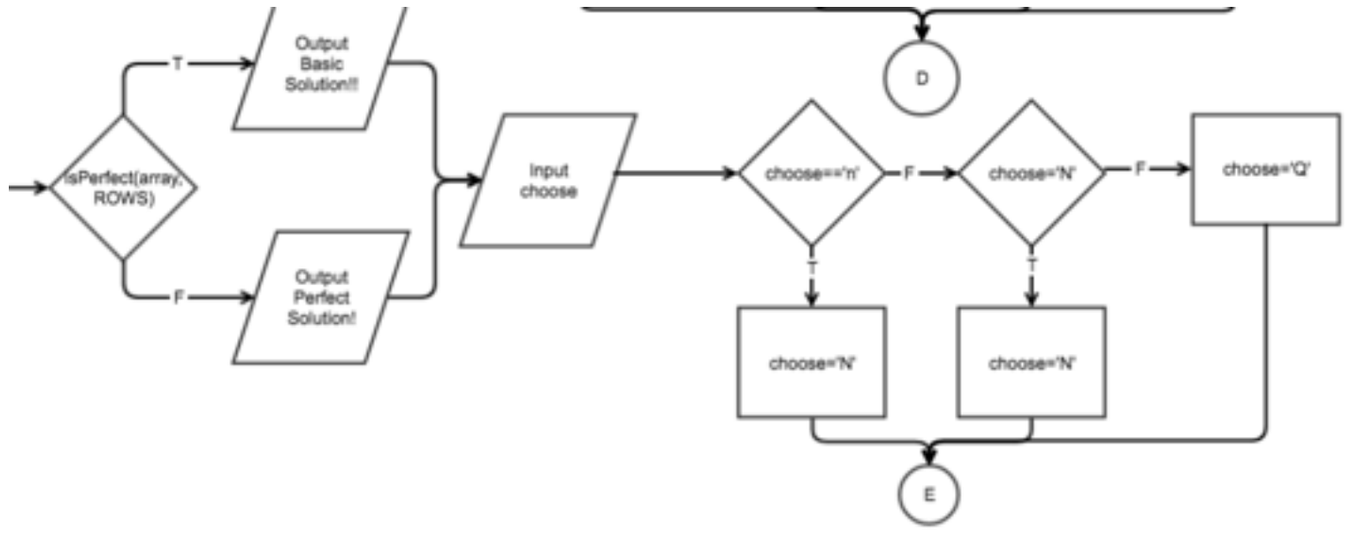


Back Menu

Then call function isFinish function determine the game is finish or not. if not, then using the while loop continue to above get position, get number etc..

if it is finished, then call function check success or not.

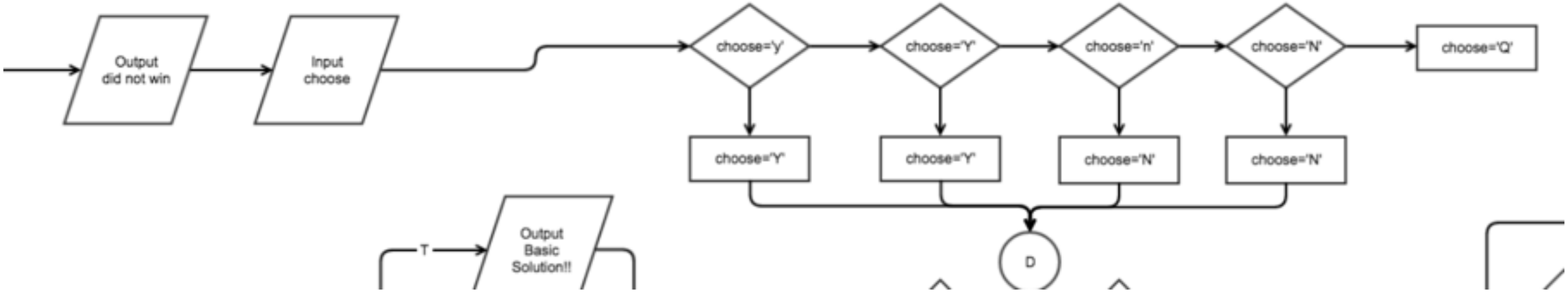




For success, check the file opening or not, and display the answer array.

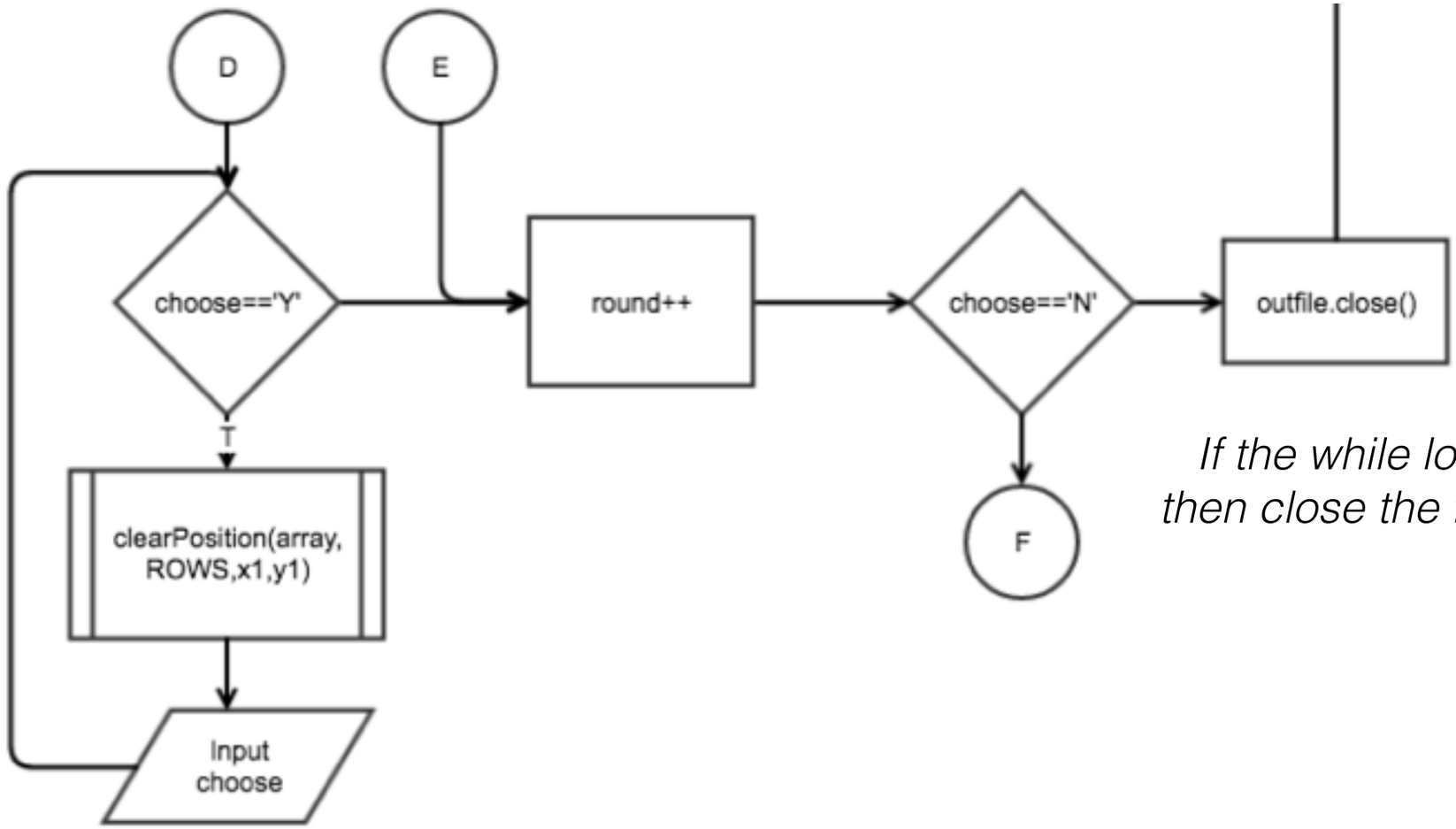
Then call function check it perfect solution or not output the result. Then the program ask the choice and switch case formatting the choice. if choice is N, in a while loop, will go to above set the array is null. if not then return 0 and exit the program.

Back Menu



If not win, output the result, then ask for choice, by using switch case formatting the choice. At here, player can go back to clear one position each time, and redo it. Or start a new round in while loop.

And also can default choice to quit the game.



If the while loop finish,
then close the file for output

Back Menu


```
int binarySearch(vector<int> &id, int size, int value){
    int first=0,          //First array element
        last=size-1,      //Last array element
        middle,           //midpoint of search
        position=-1;      //Position of search value
    bool found=false;     //Flag
    do{
        middle=(first+last)/2; //Calculate midpoint
        if(id[middle]==value){
            found=true;
            position=middle;
        }else if(id[middle]>value) last=middle-1;
        else first=middle+1;
    }while(!found && first<=last);
    return position;
}
```

binarySearch

Receive the integer vector id and its size and value for search.

set the initial first and last position.

set the found false flag.

then using do while get its middle position.

compare middle value with searching value.

if it is, then true flag, and position is middle.

if not, then check middle value greater than searching value or not.

if it dose, then last=middle-1

if not, then first=middle+1

the condition for loop is searching until the value is found or first<=last. No element to search.

then return the position.

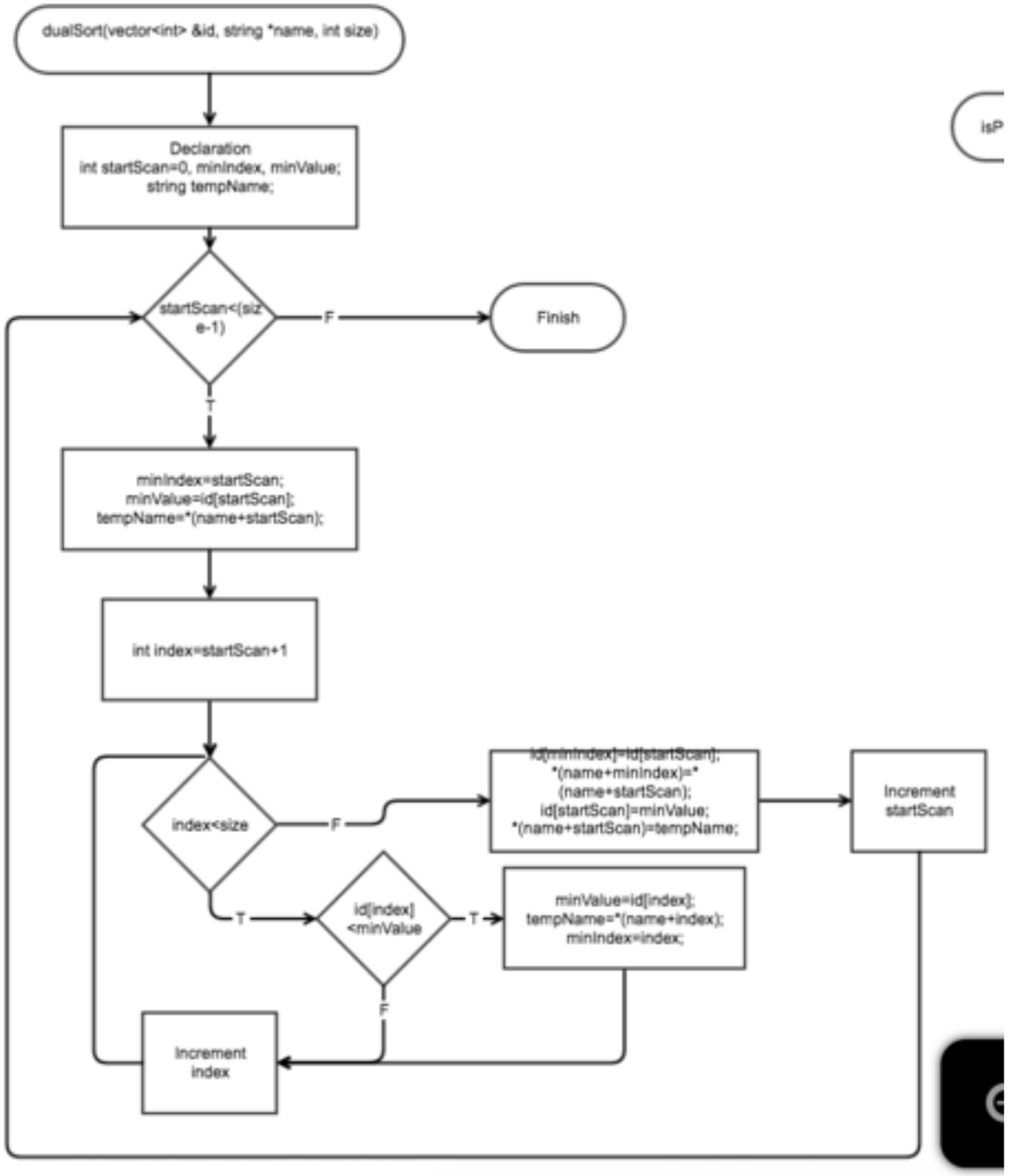
Set the initial position -1,

when the value is not found,

in the main function can recognize -1

then display it is not found in data.

Back Menu




```
void dualSort(vector<int> &id, string *name, int size){
    //Declaration the Variables
    int startScan, minIndex, minValue;
    string tempName;
    for(startScan=0; startScan<(size-1); startScan++){
        minIndex=startScan;
        minValue=id[startScan];
        tempName=*(name+startScan);
        for(int index=startScan+1; index<size; index++){
            if(id[index]<minValue){
                minValue=id[index];
                tempName=*(name+index);
                minIndex=index;
            }
        }
        id[minIndex]=id[startScan];
        *(name+minIndex)=*(name+startScan);
        id[startScan]=minValue;
        *(name+startScan)=tempName;
    }
}
```

dualSort

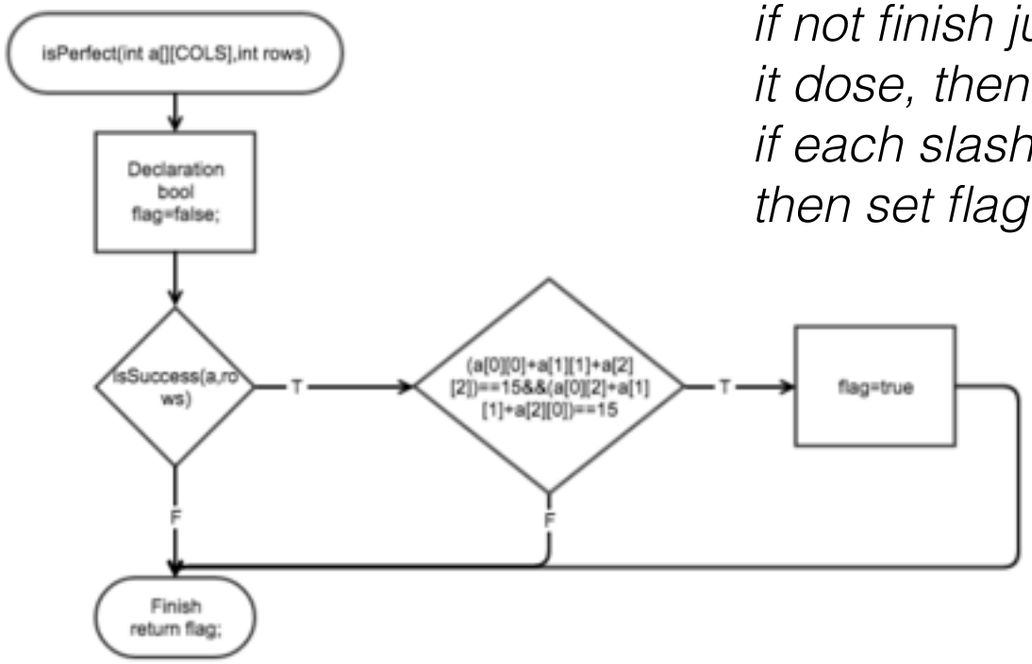
Receive the integer vector id and string name and their size.

Declare the date type. And using for loop.
set minIndex is the position declare in loop begin with 0
minValue is the value in position startScan.
tempName is name in array at position startScan
then using another for loop to compare the id number with later one. if later one smaller, then swap them.
until get the smallest one and move it to first.
then do same thing for second one, etc..

Back Menu

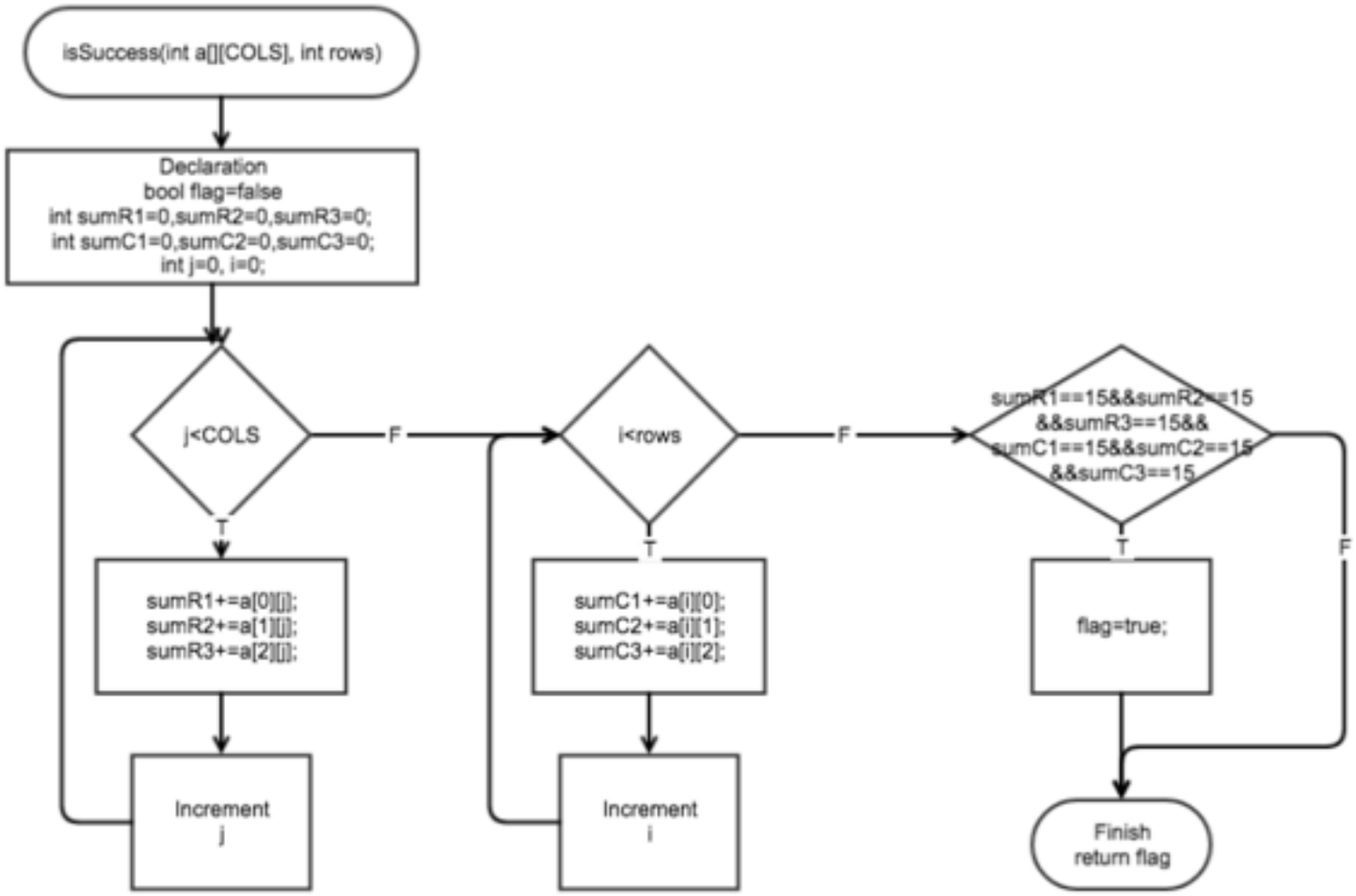
isPerfect

set the boolean flag false, then using if statement
verify the answer is success or not, based on the answer is finish
if not finish just return the false flag
it dose, then do below
if each slash in array is equal 15 at same time,
then set flag to true. And return the flag




```
bool isPerfect(int a[][COLS],int rows){
    bool flag=false;    //Set the initialize flag to false
    if(isSuccess(a,rows)){
        if((a[0][0]+a[1][1]+a[2][2])==15&&(a[0][2]+a[1][1]+a[2][0])==
            flag=true;
    }
    //Finish and return the flag
    return flag;
}
```

Back Menu




```
bool isSuccess(int a[][COLS], int rows){
    //Declaration the Variables
    bool flag=false; //Set the flag
    int sumR1=0,sumR2=0,sumR3=0; //The sum of each row
    int sumC1=0,sumC2=0,sumC3=0; //The sum of each column
    //Process
    for(int j=0; j<COLS; j++){
        sumR1+=a[0][j];
        sumR2+=a[1][j];
        sumR3+=a[2][j];
    }
    for(int i=0; i<rows; i++){
        sumC1+=a[i][0];
        sumC2+=a[i][1];
        sumC3+=a[i][2];
    }
    //Determine the sum equal 15 or not
    if(sumR1==15&&sumR2==15&&sumR3==15&&
            sumC1==15&&sumC2==15&&sumC3==15) flag=true;
    //Finish and return the flag
    return flag;
}
```

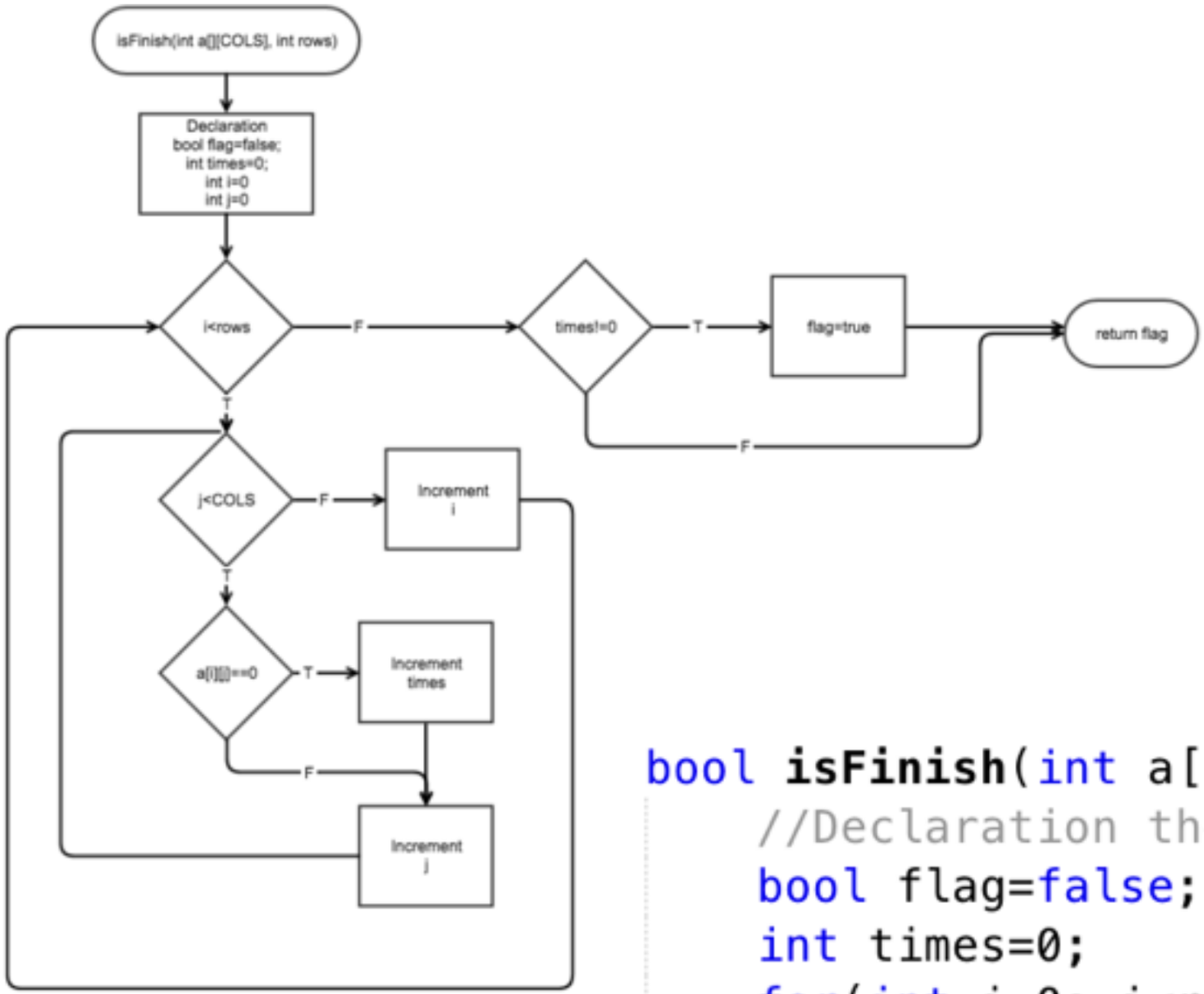
isSuccess

Set the initial flag to false, then declare sum of each row and each column then get the sum of that by using two for loop.

then determine that each sum equal 15 or not. the relationship between them is and. Therefore, it must all of sum equal 15 at same time.

For anyone is not 15, then the flag is false, the answer is not correct. then return the flag to main function,

Back Menu



isFinish

this function receive the answer array, and its row size.

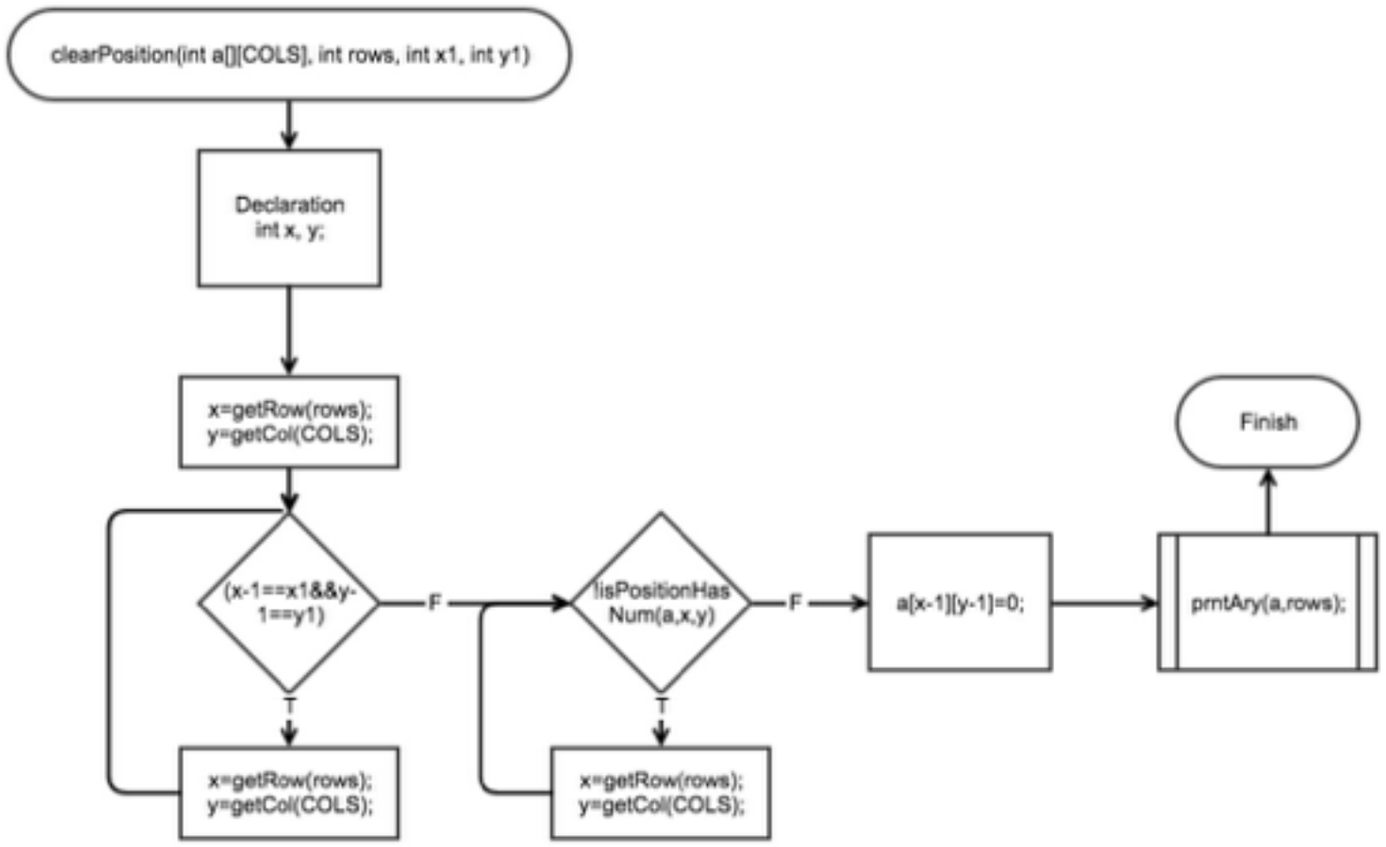
Then the function will check that all blank is fill out or not.

set the boolean flag to false. then declare times 0. use two for loop check each position if there exit 0 then increment times.

then use the if statement
if the times not equal 0
that means there still have 0 in array
is not finish
and return flag to main
main function if - statement !(isfinish)

```
bool isFinish(int a[][COLS], int rows){
    //Declaration the Variables
    bool flag=false; //Set the flag
    int times=0;     //Set the time holdi
    for(int i=0; i<rows; i++){
        for(int j=0; j<COLS; j++){
            if(a[i][j]==0) times++;
        }
    }
    if(times!=0) flag=true;
    //Finish and return flag
    return flag;
}
```

Back Menu




```
void clearPosition(int a[][COLS], int rows, int x1, int y1){
    //Declaration the Variables
    int x, y;
    //Process
    //Get the position want to clear
    x=getRow(rows);
    y=getCol(COLS);
    //Limit the original number which can not be cleared.
    while((x-1==x1&&y-1==y1)){
        cout<<"The question number can not be cleared."<<endl;
        cout<<"Please re-enter in the position."<<endl;
        x=getRow(rows);
        y=getCol(COLS);
    }
    //Check the position whether has a number
    while(!isPositionHasNum(a,x,y)){
        //if the position does not have a number
        cout<<"This position ("<<x<<","<<y
                <<") does not have a number need to clear"<<endl;
        cout<<"Please re-enter the position you want to clear."<<end
        x=getRow(rows);
        y=getCol(COLS);
    }
    //Set the position to 0
    a[x-1][y-1]=0;
    //Print out the new array
    prntAry(a,rows);
}
```

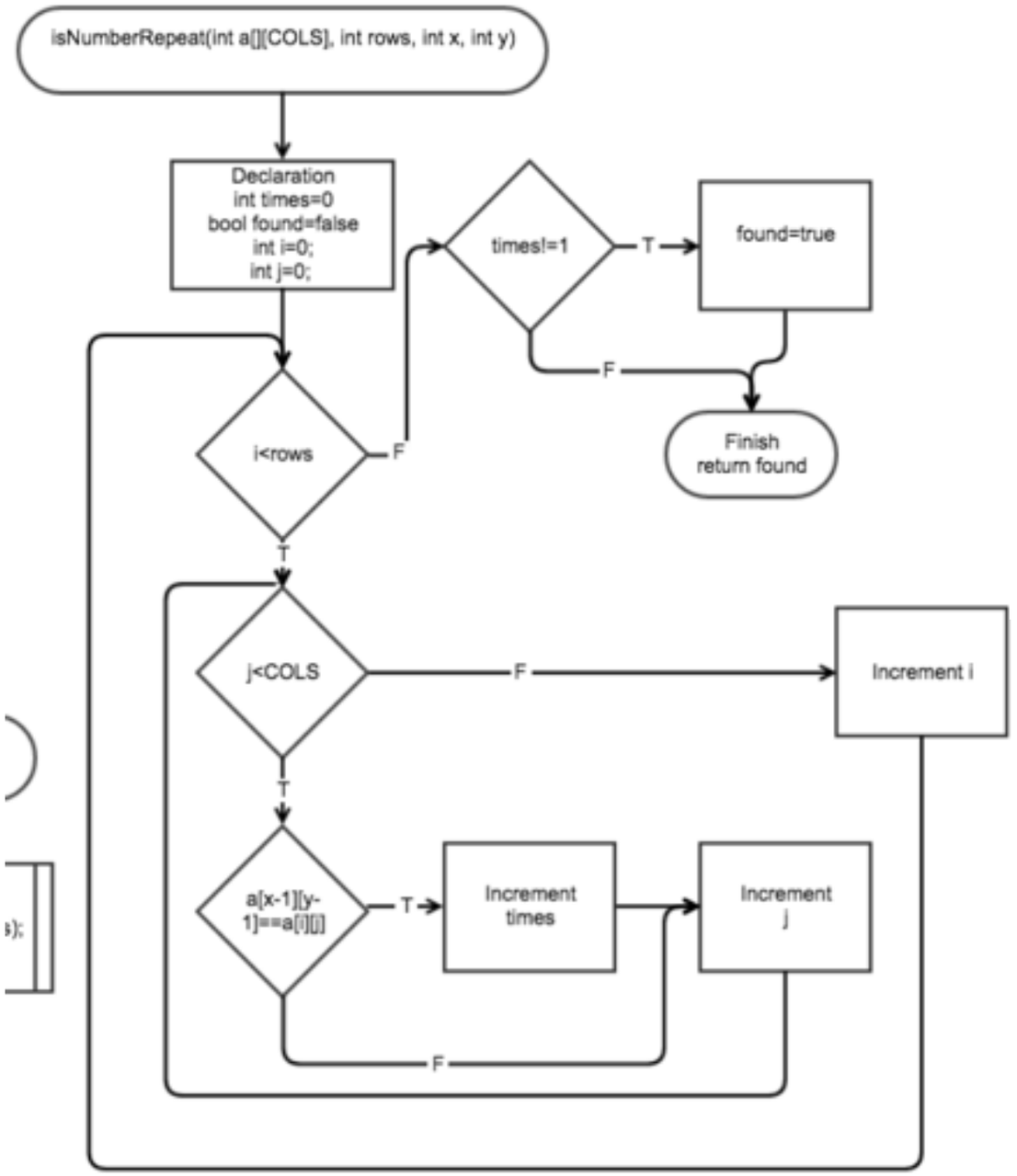
clearPosition

The function clearPosition will receives answer array, its row size and the position. then get the position the player want to clear. Determine that the position is not the original one. if it same as original then get the do while loop to re-enter in a position.

then call the function isPositionHasNum, if has a number then set this position 0, if not have a number , re-enter a position.

Finally, print out the array for display.

Back Menu




```
bool isNumberRepeat(int a[][COLS], int rows, int x, int y){
    //Declaration the variables
    int times=0; //The times one number displays
    bool found=false; //Set the flag
    for(int i=0; i<rows; i++){
        for(int j=0; j<COLS; j++){
                if(a[x-1][y-1]==a[i][j]){
                    times++;
            }
        }
    }if(times!=1) found=true;
    //Finish and return the found
    return found;
}
```

isNumberRepeat

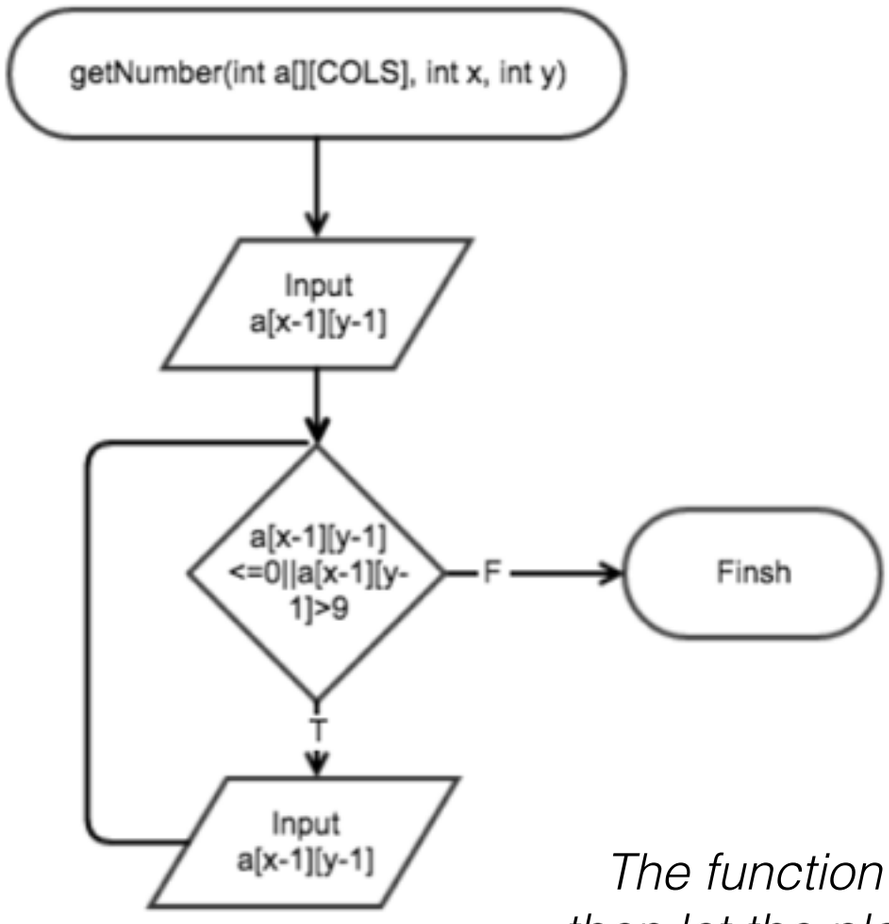
This function will check the number which the player enter in is only using once or not.

Receive the array, its row size and position, then, set the initial found= false.

declare the times=0. use two for loop check each position. if this number the user enter in equal to any element in the array. then increment the times. finally, use the if statement determine time!=1. if time not equal 1 then change found to true.

For the number enter in, this array is already has times=1. if there are two or more number same as the number user enter in, then increment the times. the times will greater than1. then flag=true. it means this number is repeat.

Back Menu

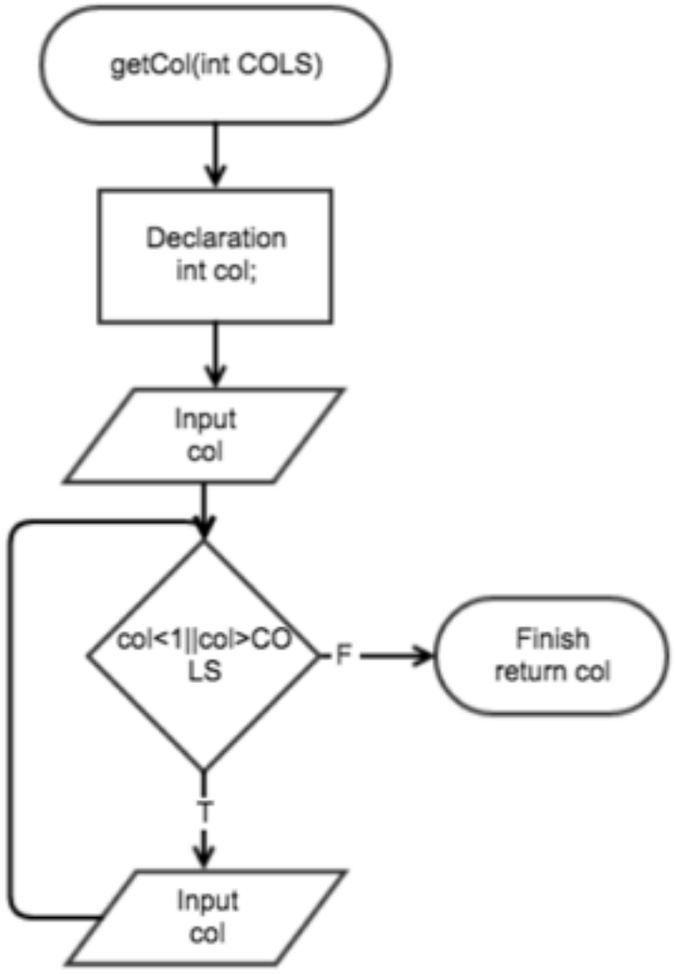


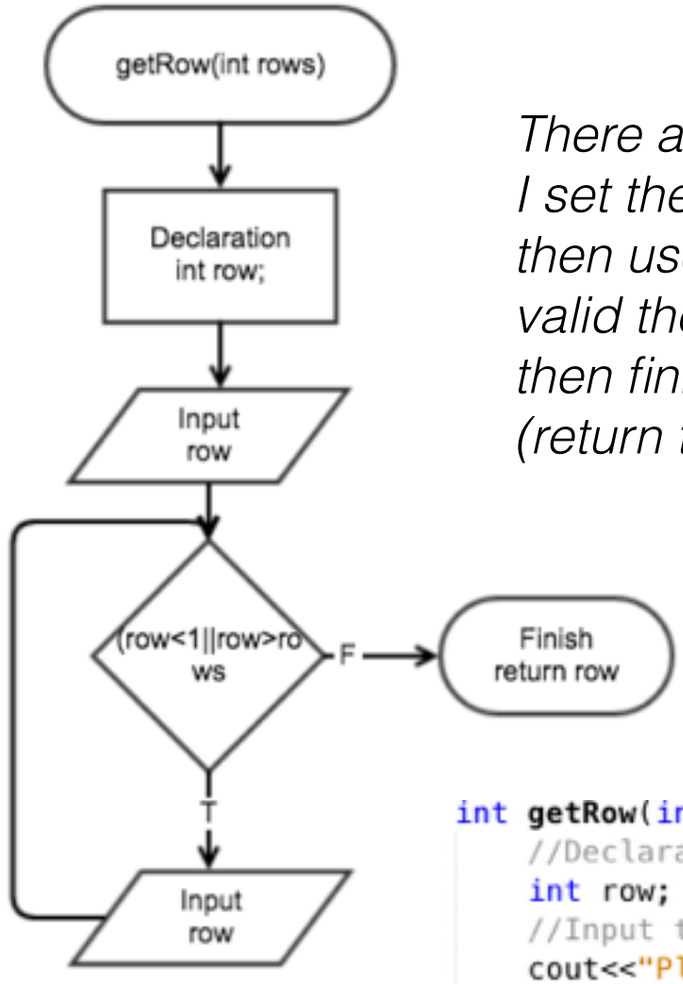
getNumber

The function get number will receive the array and position then let the player enter in a number. Valid the number between 1-9 by using while loop.

```
void getNumber(int a[][COLS], int x, int y){
        cout<<"Please enter in a number you want
        cin>>a[x-1][y-1];
    while(a[x-1][y-1]<=0||a[x-1][y-1]>9){
        cout<<"Wrong number, re-enter : ";
        cin>>a[x-1][y-1];
    }
}
```

Back Menu



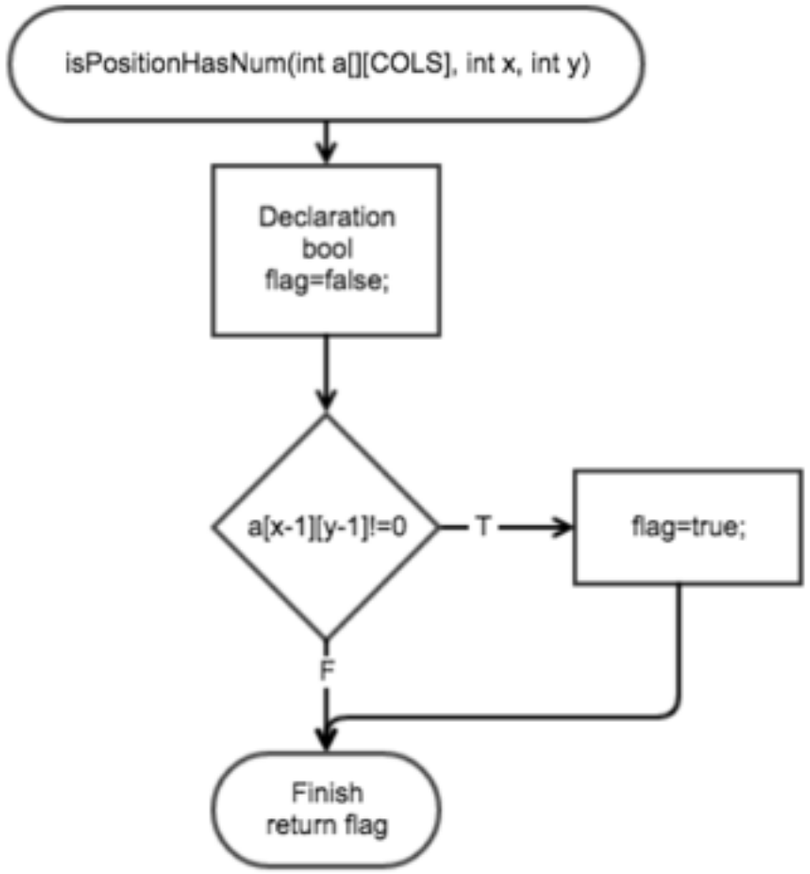


There are two function for getting in a position
I set the position formatting like (row, column)
then use while loop get the row and column.
valid the number between 1-3.
then finish and return the position.
(return the column and row)

```
int getCol(int COLS){
    //Declaration the Variables
    int col; //The number of column
    //Input the column
    cout<<"Please enter in the number of column : ";
    cin>>col;
    //check the range
    while(col<1||col>COLS){
        cout<<"Wrong number for column, re-enter in numbe
        cin>>col;
    }
    //Return the value of column
    return col;
}
```

```
int getRow(int rows){
    //Declaration the Variables
    int row; //The number of row
    //Input the number of row
    cout<<"Please enter in the number of row : ";
    cin>>row;
    //check the range
    while(row<1||row>rows){
        cout<<"Wrong number for row, re-enter in number (1,2,3):
        cin>>row;
    }
    //Return the value of row
    return row;
}
```

Back Menu

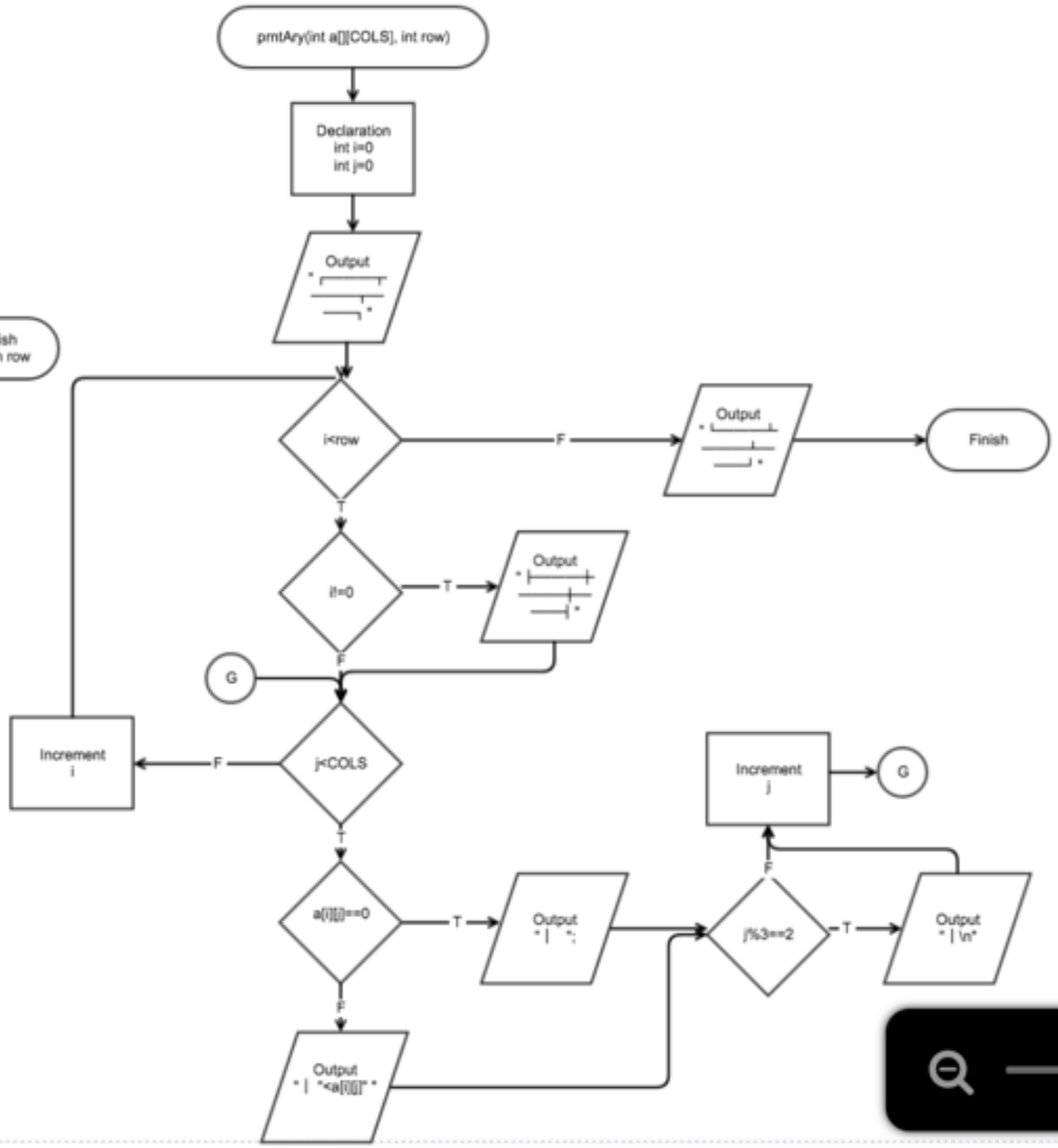


isPositionHasNum

the function isPositionHasNum receive the array and position, then set the initial flag to false.
use if statement for check the position value equal 0 or not.
b/c I set the array is null. so expect the original one, other place the user didn't enter in should be 0.
so if the position value is 0, it means the position is no num.
if there not 0, then set the flag=true. return the flag to main function

```
bool isPositionHasNum(int a[][COLS], int x, int y){
    bool flag=false;  //Set the flag holding true and false
    //if the position is not 0, then the position has a number
    if(a[x-1][y-1]!=0) flag=true;
    //Finish and return flag
    return flag;
}
```

Back Menu




```
void prntAry(int a[][COLS], int row){
    //display the array
    cout<<"┌──┬──┬──┐"<<endl;
    for(int i=0; i<row; i++){
        if(i!=0) cout<<"├──┼──┼──┤"<<endl;
        for(int j=0; j<COLS; j++){
            if(a[i][j]==0) cout<<"|   ";
            else cout<<"| "<<a[i][j]<<" ";
            if(j%3==2) cout<<"|"<<endl;
        }
    }cout<<"└──┴──┴──┘"<<endl;
    cout<<endl;
}
```

prntAry

The function prntAry main display the answer array and the chart.
Firstly, output the header.
then use for loop to loop each row.
if row not equal 0, cout the middle chart.
then use another for loop to loop each column.
if the position has no number then display a blank
if the position has a number then display the number
finally, display the footer.

Back Menu

4. Flowchart and Reference

Flowchart:

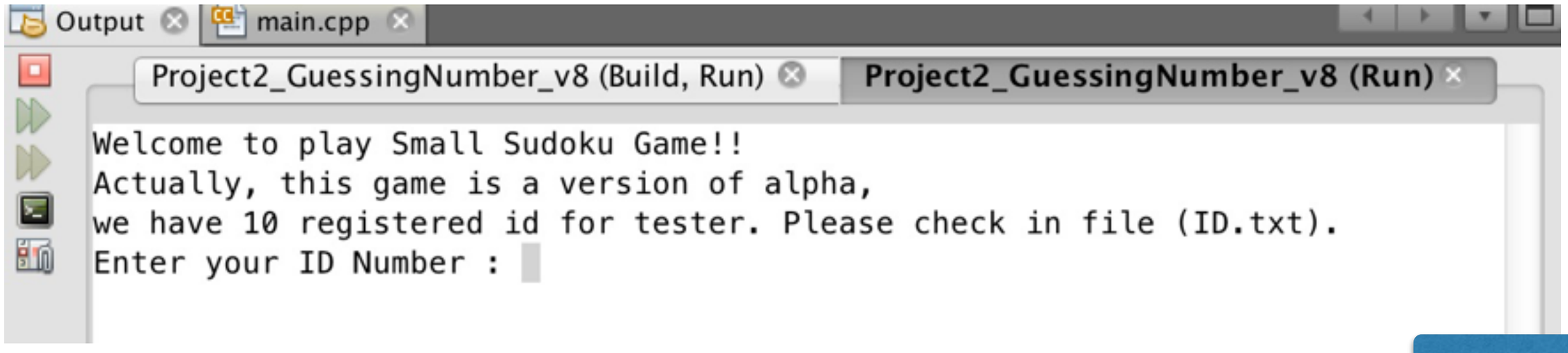
Click for Project 2 Flowchart

(by using gliffy.com)

Reference:

Gaddis 8th Edition.
Github of Professor Mark lehr

5. Game Play (Sample I/O)



Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Projects | Files | Services

- Project2_GuessingNumber_v8
  - Header Files
  - Resource Files
  - Source Files
    - main.cpp
  - Test Files
  - Important Files

isNumberRepeat(int a[][], int rows...

- COLS
- binarySearch(vector<int>& id, int
- clearPosition(int a[][], int rows, int
- displayMessage(string Name)
- dualSort(vector<int>& id, string*
- getCol(int COLS)
- getNumber(int a[][], int x, int y)
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][], int rows
- isPerfect(int a[][], int rows)
- isPositionHasNum(int a[][], int x, i
- isSuccess(int a[][], int rows)
- main(int argc, char** argv)
- prntAry(int a[][], int row)

Output | main.cpp

Project2_GuessingNumber_v8 (Build, Run) | Project2_GuessingNumber_v8 (Run)

```
Welcome to play Small Sudoku Game!!
Actually, this game is a version of alpha,
we have 10 registered id for tester. Please check in file (ID.txt).
Enter your ID Number : 313
Hi, Johnson Jill. Let me introduce you for the rule of this game:
  1. You will get a 3x3 chart which already had a random number in random p
osition.
  2. You need to use 1-9 (each number only can use once) fill out
     all other blank position using (row,colum).
  3. You must make it to the sum of each row and each column
     equal to 15 to get basic win, and if you can make a perfect
     answer which the sum of each slash equal 15 also.
  4. You can clear the position (expect original one) and enter again.

Okay, Johnson Jill. Are you ready to challenge yourself?
Enter any key to go!!
```

Project2_GuessingNumber_v8 (Run)

463

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Projects | Files | Services

- Project2_GuessingNumber_v8
  - Header Files
  - Resource Files
  - Source Files
    - main.cpp
  - Test Files
  - Important Files

isNumberRepeat(int a[][], int rows...

- COLS
- binarySearch(vector<int>& id, int
- clearPosition(int a[][], int rows, int
- displayMessage(string Name)
- dualSort(vector<int>& id, string*
- getCol(int COLS)
- getNumber(int a[][], int x, int y)
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][], int rows
- isPerfect(int a[][], int rows)
- isPositionHasNum(int a[][], int x, i
- isSuccess(int a[][], int rows)
- main(int argc, char** argv)
- prntAry(int a[][], int row)

Output | main.cpp

Project2_GuessingNumber_v8 (Build, Run) | Project2_GuessingNumber_v8 (Run)

```
Welcome to play Small Sudoku Game!!
Actually, this game is a version of alpha,
we have 10 registered id for tester. Please check in file (ID.txt).
Enter your ID Number : 313
Hi, Johnson Jill. Let me introduce you for the rule of this game:
  1. You will get a 3x3 chart which already had a random number in random p
osition.
  2. You need to use 1-9 (each number only can use once) fill out
     all other blank position using (row,colum).
  3. You must make it to the sum of each row and each column
     equal to 15 to get basic win, and if you can make a perfect
     answer which the sum of each slash equal 15 also.
  4. You can clear the position (expect original one) and enter again.

Okay, Johnson Jill. Are you ready to challenge yourself?
Enter any key to go!!

Problem Challenge:
```

8		

<- Here is the original number in question

```
Which row you want to put a number in?
Please enter in the number of row :
```

Project2_GuessingNumber_v8 (Run)

463

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Projects | Files | Services

- Project2_GuessingNumber_v8
  - Header Files
  - Resource Files
  - Source Files
    - main.cpp
  - Test Files
  - Important Files

isNumberRepeat(int a[][], int rows...

- COLS
- binarySearch(vector<int>& id, int
- clearPosition(int a[][], int rows, int
- displayMessage(string Name)
- dualSort(vector<int>& id, string*
- getCol(int COLS)
- getNumber(int a[][], int x, int y)
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][], int rows
- isPerfect(int a[][], int rows)
- isPositionHasNum(int a[][], int x, i
- isSuccess(int a[][], int rows)
- main(int argc, char** argv)
- prntAry(int a[][], int row)

Output | main.cpp

Project2_GuessingNumber_v8 (Build, Run) | Project2_GuessingNumber_v8 (Run)

8		

```
Which row you want to put a number in?
Please enter in the number of row : 1
Which column you want to put a number in?
Please enter in the number of column : 2
Please enter in a number you want put in (1,2) : 1
```

<- enter the position

<- enter the value

8	1	

```
Type Y to clear one position, or any key to continue: y
Please enter in the number of row : 1
Please enter in the number of column : 2
```

<- for clear the position

8		

```
Type Y to clear one position, or any key to continue:
```

Project2_GuessingNumber_v8 (Run)

463:

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Projects | Files | Services

- Project2_GuessingNumber_v8
  - Header Files
  - Resource Files
  - Source Files
    - main.cpp
  - Test Files
  - Important Files

isNumberRepeat(int a[][], int rows...

- COLS
- binarySearch(vector<int>& id, int
- clearPosition(int a[][], int rows, int
- displayMessage(string Name)
- dualSort(vector<int>& id, string*
- getCol(int COLS)
- getNumber(int a[][], int x, int y)
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][], int rows
- isPerfect(int a[][], int rows)
- isPositionHasNum(int a[][], int x, i
- isSuccess(int a[][], int rows)
- main(int argc, char** argv)
- prntAry(int a[][], int row)

Output | main.cpp

Project2_GuessingNumber_v8 (Build, Run) | Project2_GuessingNumber_v8 (Run)


```
Type Y to clear one position, or any key to continue: n
```

8		

```
Which row you want to put a number in?
Please enter in the number of row : 1
Which column you want to put a number in?
Please enter in the number of column : 1
This position (1,1) already has a number.
Please enter in the number of row : 1
Please enter in the number of column : 2
Please enter in a number you want put in (1,2) : 1
```

8	1	

```
Type Y to clear one position, or any key to continue:
```

<- The player can not enter the position which already has a number

Project2_GuessingNumber_v8 (Run)

46

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Projects | Files | Services

- Project2_GuessingNumber_v8
  - Header Files
  - Resource Files
  - Source Files
    - main.cpp
  - Test Files
  - Important Files

isNumberRepeat(int a[][], int rows...

- COLS
- binarySearch(vector<int>& id, int
- clearPosition(int a[][], int rows, int
- displayMessage(string Name)
- dualSort(vector<int>& id, string*
- getCol(int COLS)
- getNumber(int a[][], int x, int y)
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][], int rows
- isPerfect(int a[][], int rows)
- isPositionHasNum(int a[][], int x, i
- isSuccess(int a[][], int rows)
- main(int argc, char** argv)
- prntAry(int a[][], int row)

Output | main.cpp

Project2_GuessingNumber_v8 (Build, Run) | Project2_GuessingNumber_v8 (Run)

8	1	6

```
Type Y to clear one position, or any key to continue: 2
```

8	1	6

```
Which row you want to put a number in?
Please enter in the number of row : 2
Which column you want to put a number in?
Please enter in the number of column : 5
Wrong number for column, re-enter in number (1,2,3): 2
Please enter in a number you want put in (2,2) : 5
```

<- The row and column must between 1-3

8	1	6
	5	

```
Type Y to clear one position, or any key to continue:
```

Project2_GuessingNumber_v8 (Run)

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Projects | Files | Services

- Project2_GuessingNumber_v8
  - Header Files
  - Resource Files
  - Source Files
    - main.cpp
  - Test Files
  - Important Files

isNumberRepeat(int a[][], int rows...

- COLS
- binarySearch(vector<int>& id, int
- clearPosition(int a[][], int rows, int
- displayMessage(string Name)
- dualSort(vector<int>& id, string*
- getCol(int COLS)
- getNumber(int a[][], int x, int y)
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][], int rows
- isPerfect(int a[][], int rows)
- isPositionHasNum(int a[][], int x, i
- isSuccess(int a[][], int rows)
- main(int argc, char** argv)
- prntAry(int a[][], int row)

Output | main.cpp

Project2_GuessingNumber_v8 (Build, Run) | Project2_GuessingNumber_v8 (Run)

3	5	7
	9	2

```
Type Y to clear one position, or any key to continue: b
```

8	1	6
3	5	7
	9	2

```
Which row you want to put a number in?
Please enter in the number of row : 3
Which column you want to put a number in?
Please enter in the number of column : 1
Please enter in a number you want put in (3,1) : 4
```

8	1	6
3	5	7
4	9	2

```
Type Y to clear one position, or any key to continue: n

YOU ARE VERY SMART, YOU MADE A PERFECT SOLUTION!!
Type N to start a new round, or Q to quit this game:
```

Project2_GuessingNumber_v8 (Run)

463:5

Back Menu

records.txt

round 1

8	1	6
3	5	7
4	9	2

by Johnson Jill ID: 313
Perfect Solution!

ID.txt

158
323
127
996
834
226
289
313
167
728

The winning result will be output in record.txt file.

And here are 10 registered ID for tester in ID.txt file

Back Menu

6. Program Code

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

prntAry(int a[][], int row...

- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[]|
- isSuccess(int a[][], int rov
- main(int argc, char** arg
- prntAry(int a[][], int row)

main.cpp

Source | History

```
1  /*
2   * File:   main.cpp
3   * Author: Weikang Du
4   * Created on November 29, 2016, 11:48 AM
5   * Purpose: Making a Small Sudoku Game.
6   */
7
8  #include <iostream> //I/O
9  #include <ctime>    //Time
10 #include <cstdlib>  //Random
11 #include <fstream>  //I/O file
12 #include <string>   //String name
13 #include <vector>   //Vector
14
15 using namespace std;
16
17 //User Libraries Here
18
19 //Global Constants Only, No Global Variables
20 //Like PI, e, Gravity, or conversions
21 const int COLS=3;      //Set the array columns is 3
22 //Declaration Alpha structure holds information about tester
23 struct Alpha{
24     int ID;          //Integer ID holds the tester's id
25     string Name;     //String Name holds the tester's name
26 };
27 //Function Prototypes Here
28 void prntAry(int [][COLS], int);
29 int getRow(int);
30 int getCol(int COLS);
31 bool isPositionHasNum(int [][COLS], int, int);
32 void getNumber(int [][COLS], int, int);
33 bool isNumberRepeat(int [][COLS], int, int, int);
```

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

Alpha - Navigator

Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[]
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
32      void getNumber(int [][COLS], int, int);
33      bool isNumberRepeat(int [][COLS], int, int, int);
34      void clearPosition(int [][COLS], int, int, int);
35      bool isFinish(int [][COLS], int);
36      bool isSuccess(int [][COLS], int);
37      bool isPerfect(int [][COLS],int);
38      void dualSort(vector<int> &, string*, int);
39      int binarySearch(vector<int> &, int, int);
40      void displayMessage(string);
41      //Program Execution Begins Here!
42      int main(int argc, char** argv) {
43          //Initialize the random number seed
            srand(static_cast<unsigned int>(time(0)));
45          //Declare all Variables here
46          const int ROWS=3;    //Set the array rows is 3
47          int x, y;            //The position row and column
48          char choice, choose; //The variables holding user's choice
49          int round=1;         //The number of round.
50          ofstream outfile;    //Output file
51          ifstream infile;     //input file
52          const int NUM=10;    //The size of idNum and Name array
53          vector<int> idNum(NUM);      //The vector holding ID number
54          string Name[NUM]={"Collins Bill", "Smith Bart", "Allen Jim", "Griffin, Jim",
55                            "Smith Marty", "Rose Gary", "Taylor Swith", "Johnson Jill"
56                           ,"Allison Jeff", "Looney Joe"};
57          Alpha tester;        //Declaration the structure variable
58          int position;        //The variable holding the position of id and name array
59          //Open the file
            outfile.open("records.txt");
            infile.open("ID.txt");
62          //read the file to vector
63          if(!infile){
64              cout<<"Open file failed!"<<endl;
```

33:28 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...
- Alpha
- COLS
- binarySearch(vector<int:
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[]
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
62      //read the file to vector
63      if(!infile){
            cout<<"Open file failed!"<<endl;
65          exit(0);
66      }else{
67          for(int i=0; i<NUM; i++){
68              infile>>idNum[i];
69          }
70      }
71      //close the file
        infile.close();
73      //Sort the array
74      dualSort(idNum, Name, NUM);
75      //display the message
76      cout<<"Welcome to play Small Sudoku Game!!"<<endl;
77      cout<<"Actually, this game is a version of alpha, "<<endl;
78      cout<<"we have 10 registered id for tester. Please check in file (ID.txt)."<<endl;
79      cout<<"Enter your ID Number : ";
80      cin>>tester.ID;
81      //Determine the ID is valid or not
82      while(tester.ID<100||tester.ID>999){
83          cout<<"Wrong ID, it must between 100 to 999, re-enter: ";
84          cin>>tester.ID;
85      }
86      //Determine that the ID exits or not.
87      while(binarySearch(idNum, NUM, tester.ID)==-1){
88          cout<<"The ID is invalid, please check ID in 'ID.txt'. Re-enter: ";
89          cin>>tester.ID;
90      }
91      position=binarySearch(idNum, NUM, tester.ID);
92      tester.Name=Name[position];
93      //Display message
94      displayMessage(tester.Name);
```

64:10 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[]|
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
95      //Processing here!!
96      do{
97          //fill the array
98          int array[ROWS][COLS]={}; //Set the two dimension array and
99                                    //make the elements of array to 0.
100         //Declaration the Variables
101         int x1,y1;          //random position x and y;
102         //Process
            x1=rand()%3;
            y1=rand()%3;
105         //Determine
106         //fill the array
            array[x1][y1]=rand()%9+1;
108         //Process here!
109         do{
110             //display the array
111             prntAry(array, ROWS);
112             //Get the number of row for entering number
113             cout<<"Which row you want to put a number in?"<<endl;
114             x=getRow(ROWS);
115             //Get the number of column for entering number.
116             cout<<"Which column you want to put a number in?"<<endl;
117             y=getCol(COLS);
118             //check the position whether has a number
119             while(isPositionHasNum(array,x,y)){
120                 cout<<"This position ("<<x<<","<<y<<") already has a number."<<endl;
121                 x=getRow(ROWS);
122                 y=getCol(COLS);
123             }
124             //Put the number in array
125             getNumber(array,x,y);
126             //Display the array
127             prntAry(array, ROWS);
```

94:17 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[][
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
128             //Check the number whether already has;
129             while(isNumberRepeat(array,ROWS,x,y)){
130                 cout<<"You only can use this number once."<<endl;
131                 //Clear this position to 0;
132                 array[x-1][y-1]=0;
133                 //Print out the array
134                 prntAry(array, ROWS);
135                 //Re-enter in the number
136                 getNumber(array,x,y);
137                 //Print out the array
138                 prntAry(array, ROWS);
139             }
140             //Determine the choice to clear position.
141             cout<<"Type Y to clear one position, or any key to continue: ";
142             cin>>choice;
143             while(choice=='Y'||choice=='y'){
144                 clearPosition(array, ROWS,x1,y1);
145                 cout<<"Type Y to clear one position, or any key to continue: ";
146                 cin>>choice;
147             }
148         }while(isFinish(array,ROWS));    //Check whether game is finished.
149
150         //Check whether user is win or not.
151         if(isSuccess(array, ROWS)){
152             if(!outfile){
153                 cout<<"Open file failed!"<<endl;
154                 exit(0);
155             }
156             else{
157                 outfile<<"***round "<<round<<"***"<<endl;
158                 outfile<<"┌─┬─┬─┐"<<endl;
159                 for(int i=0; i<ROWS; i++){
160                     if(i!=0) outfile<<"├─┼─┼─┤"<<endl;
```

127:29 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[]
- isSuccess(int a[][], int rov
- main(int argc, char** arg

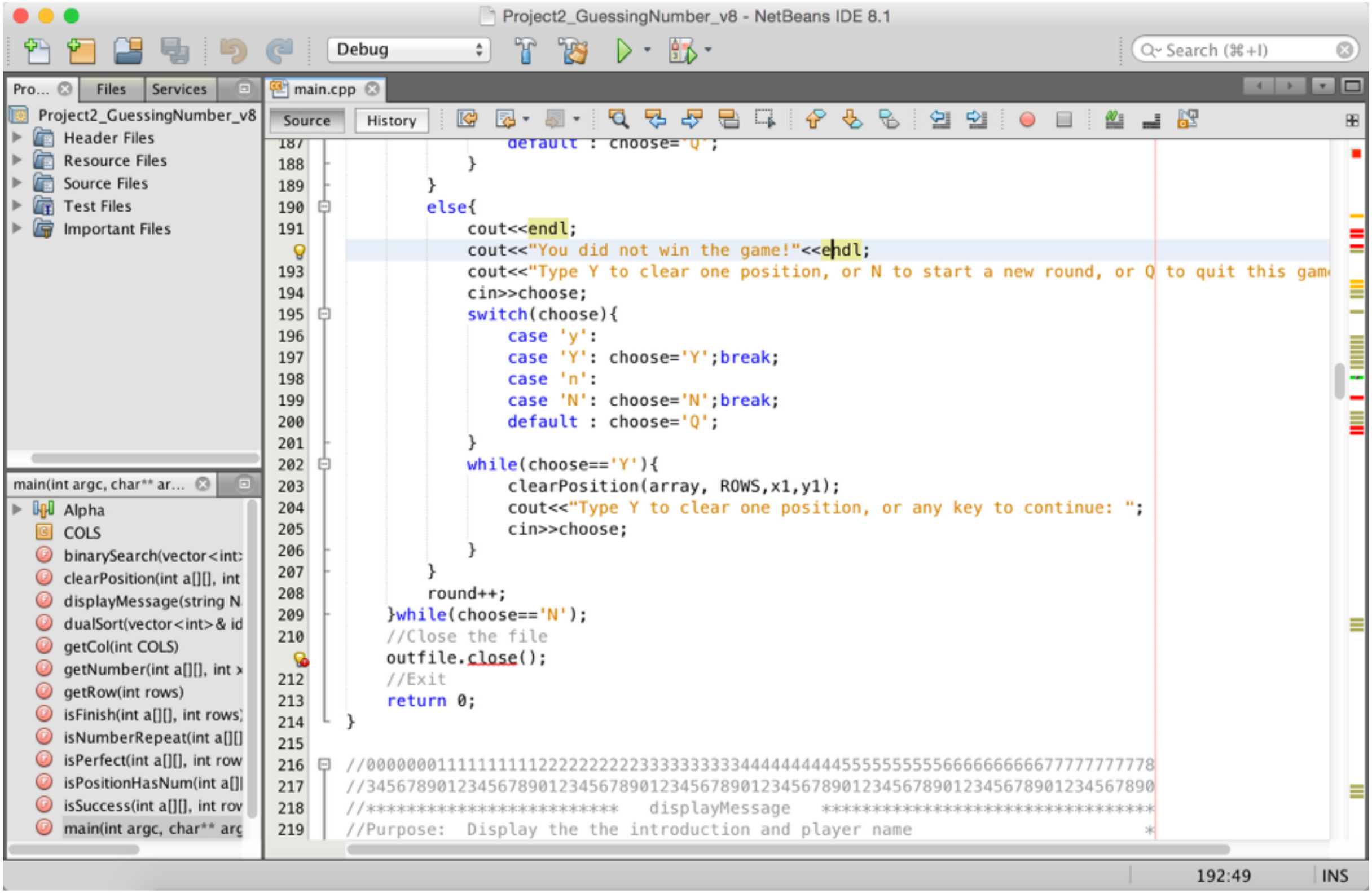
main.cpp

Source | History

```
161                 for(int j=0; j<COLS; j++){
162                     outfile<<"| "<<array[i][j]<<" ";
163                     if(j%3==2) outfile<<"|"<<endl;
164                 }
165             }outfile<<"└──┴──┴──┘"<<endl;
166             outfile<<"\tby "<<tester.Name<<" ID: "<<tester.ID<<endl;
167         }
168
169         //Check the answer is perfect or not.
170         if(isPerfect(array, ROWS)){
171             cout<<endl;
172             cout<<"YOU ARE VERY SMART, YOU MADE A PERFECT SOLUTION!!"<<endl;
173             outfile<<"Perfect Solution!"<<endl;
174             outfile<<endl;
175         }else{
176             cout<<endl;
177             cout<<"CONGRADUATIONS!! YOU WON THE GAME!"<<endl;
178             outfile<<"Basic Solution!!"<<endl;
179             outfile<<endl;
180         }
181
182         cout<<"Type N to start a new round, or Q to quit this game:"<<endl;
183         cin>>choose;
184         switch(choose){
185             case 'n':
186             case 'N': choose='N';break;
187             default : choose='Q';
188         }
189     }
190     else{
191         cout<<endl;
192         cout<<"You did not win the game!"<<endl;
```

160:33 | INS

Back Menu


Project2_GuessingNumber_v8 - NetBeans IDE 8.1
Debug
Search (⌘+I)
Pro...
Files
Services
Project2_GuessingNumber_v8
Header Files
Resource Files
Source Files
Test Files
Important Files
main(int argc, char** ar...
Alpha
COLS
binarySearch(vector<int>
clearPosition(int a[][], int
displayMessage(string N
dualSort(vector<int>& id
getCol(int COLS)
getNumber(int a[][], int x
getRow(int rows)
isFinish(int a[][], int rows)
isNumberRepeat(int a[][]
isPerfect(int a[][], int row
isPositionHasNum(int a[]
isSuccess(int a[][], int rov
main(int argc, char** arg
main.cpp
Source
History
default : choose='Q';
}
}
else{
cout<<endl;
cout<<"You did not win the game!"<<endl;
cout<<"Type Y to clear one position, or N to start a new round, or Q to quit this game
cin>>choose;
switch(choose){
case 'y':
case 'Y': choose='Y';break;
case 'n':
case 'N': choose='N';break;
default : choose='Q';
}
while(choose=='Y'){
clearPosition(array, ROWS,x1,y1);
cout<<"Type Y to clear one position, or any key to continue: ";
cin>>choose;
}
}
round++;
}while(choose=='N');
//Close the file
outfile.close();
//Exit
return 0;
}
//0000000111111111122222222223333333333444444444455555555556666666666777777777
//3456789012345678901234567890123456789012345678901234567890123456789012345678
//************************** displayMessage ********************************
//Purpose: Display the the introduction and player name
192:49
INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[][
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
214 }
215
216 //0000000111111111122222222223333333333444444444455555555556666666666777777777778
217 //3456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
218 //**************************   displayMessage   ***********************************
219 //Purpose:  Display the the introduction and player name                            *
220 //Inputs:   Inputs to the function here -> Description, Range, Units               *
221 //  Name->The variable hold the player's name                                       *
222 //Output:   Outputs to the function here -> Description, Range, Units              *
223 //*********************************************************************************
224
225 void displayMessage(string Name){
226     cout<<"Hi, "<<Name<<". Let me introduce you for the rule of this game:"<<endl;
227     cout<<"  1. You will get a 3x3 chart which already had a random "<<
228           "number in random position."<<endl;
229     cout<<"  2. You need to use 1-9 (each number only can use once) fill out"<<endl;
230     cout<<"     all other blank position using (row,colum)."<<endl;
231     cout<<"  3. You must make it to the sum of each row and each column"<<endl;
232     cout<<"     equal to 15 to get basic win, and if you can make a perfect"<<endl;
233     cout<<"     answer which the sum of each slash equal 15 also."<<endl;
234     cout<<"  4. You can clear the position (expect original one) "<<
235           "and enter again."<<endl;
236     cout<<endl;
237     cout<<"Okay, "<<Name<<". Are you ready to challenge yourself?"<<endl;
238     cout<<"Enter any key to go!!"<<endl;
    cin.ignore();
    cin.get();
241     cout<<"Problem Challenge: "<<endl;
242     cout<<endl;
243 }
244
245 //0000000111111111122222222223333333333444444444455555555556666666666777777777778
246 //3456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

192:49 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[]
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
244
245 //0000000111111111122222222223333333333444444444455555555556666666666777777777 8
246 //3456789012345678901234567890123456789012345678901234567890123456789012345678901
247 //**************************   binarySearch   ************************************
248 //Purpose:  Search the value and its position                                     *
249 //Inputs:   Inputs to the function here -> Description, Range, Units             *
250 //  *id->The integer array hold with ID number                                   *
251 //  size->The size of id array                                                   *
252 //  value->The value which the user enter in and want to search                  *
253 //Output:   Outputs to the function here -> Description, Range, Units           *
254 //  Return the value position                                                    *
255 //*****************************************************************************
256
257 int binarySearch(vector<int> &id, int size, int value){
258     int first=0,        //First array element
259         last=size-1,    //Last array element
260         middle,         //midpoint of search
261         position=-1;    //Position of search value
262     bool found=false;   //Flag
263     do{
264         middle=(first+last)/2; //Calculate midpoint
265         if(id[middle]==value){
266             found=true;
267             position=middle;
268         }else if(id[middle]>value) last=middle-1;
269         else first=middle+1;
270     }while(!found && first<=last);
271     return position;
272 }
273
274 //0000000111111111122222222223333333333444444444455555555556666666666777777777 8
275 //3456789012345678901234567890123456789012345678901234567890123456789012345678901
276 //**************************   dualSort   ************************************
```

192:49 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[][
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
274 //0000000011111111112222222222333333333344444444445555555555666666666677777777778
275 //34567890123456789012345678901234567890123456789012345678901234567890123456789012345678901
276 //**************************   dualSort   ************************************
277 //Purpose:  Sort the id array by ascending order with its parallel name array  *
278 //Inputs:   Inputs to the function here -> Description, Range, Units           *
279 //  *id->The integer array hold with ID number                                 *
280 //  *name->The string array hold with name                                     *
281 //  size->the size of two parallel array                                       *
282 //Output:   Outputs to the function here -> Description, Range, Units          *
283 //  Saving the two array in the ascending order                                *
284 //*****************************************************************************
285
286 void dualSort(vector<int> &id, string *name, int size){
287     //Declaration the Variables
288     int startScan, minIndex, minValue;
289     string tempName;
290     for(startScan=0; startScan<(size-1); startScan++){
291         minIndex=startScan;
292         minValue=id[startScan];
293         tempName=*(name+startScan);
294         for(int index=startScan+1; index<size; index++){
295             if(id[index]<minValue){
296                 minValue=id[index];
297                 tempName=*(name+index);
298                 minIndex=index;
299             }
300         }
301         id[minIndex]=id[startScan];
302         *(name+minIndex)=*(name+startScan);
303         id[startScan]=minValue;
304         *(name+startScan)=tempName;
305     }
306 }
```

192:49 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...
- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[][
- isSuccess(int a[][], int rov
- main(int argc, char** arg

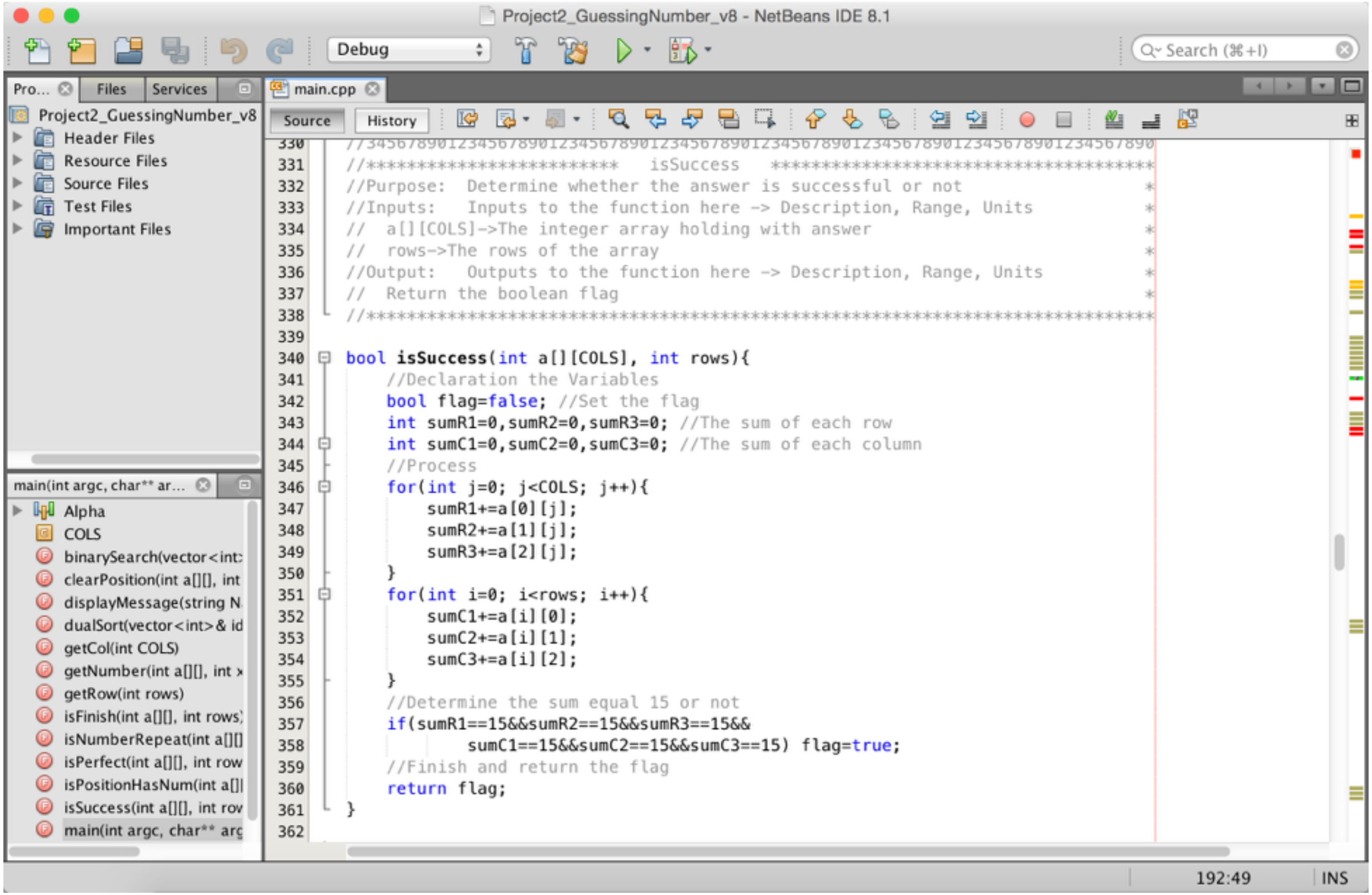
main.cpp

Source | History

```
306     }
307
308     //0000000111111111122222222223333333333444444444455555555556666666666777777777 8
309     //3456789012345678901234567890123456789012345678901234567890123456789012345678 90
310     //**************************   isPerfect   ************************************
311     //Purpose:  Determine whether the answer is perfect solution or not            *
312     //Inputs:   Inputs to the function here -> Description, Range, Units           *
313     //  a[][COLS]->The integer array holding the answer                            *
314     //  rows->the number of rows of the array                                     *
315     //Output:   Outputs to the function here -> Description, Range, Units         *
316     //  Return the boolean flag                                                   *
317     //*****************************************************************************
318
319     bool isPerfect(int a[][COLS],int rows){
320         bool flag=false;    //Set the initialize flag to false
321         if(isSuccess(a,rows)){
322             if((a[0][0]+a[1][1]+a[2][2])==15&&(a[0][2]+a[1][1]+a[2][0])==15)
323                 flag=true;
324         }
325         //Finish and return the flag
326         return flag;
327     }
328
329     //0000000111111111122222222223333333333444444444455555555556666666666777777777 8
330     //3456789012345678901234567890123456789012345678901234567890123456789012345678 90
331     //**************************   isSuccess   ************************************
332     //Purpose:  Determine whether the answer is successful or not                  *
333     //Inputs:   Inputs to the function here -> Description, Range, Units           *
334     //  a[][COLS]->The integer array holding with answer                           *
335     //  rows->The rows of the array                                               *
336     //Output:   Outputs to the function here -> Description, Range, Units         *
337     //  Return the boolean flag                                                   *
338     //*****************************************************************************
```

192:49 | INS

Back Menu


Project2_GuessingNumber_v8 - NetBeans IDE 8.1
Debug
Search (⌘+I)
Pro...
Files
Services
Project2_GuessingNumber_v8
Header Files
Resource Files
Source Files
Test Files
Important Files
main(int argc, char** ar...
Alpha
COLS
binarySearch(vector<int>
clearPosition(int a[][], int
displayMessage(string N
dualSort(vector<int>& id
getCol(int COLS)
getNumber(int a[][], int x
getRow(int rows)
isFinish(int a[][], int rows)
isNumberRepeat(int a[][]
isPerfect(int a[][], int row
isPositionHasNum(int a[]
isSuccess(int a[][], int rov
main(int argc, char** arg
main.cpp
Source
History
330 //3456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
331 //*************************** isSuccess ***************************************
332 //Purpose: Determine whether the answer is successful or not *
333 //Inputs: Inputs to the function here -> Description, Range, Units *
334 // a[][COLS]->The integer array holding with answer *
335 // rows->The rows of the array *
336 //Output: Outputs to the function here -> Description, Range, Units *
337 // Return the boolean flag *
338 //***
339
340 bool isSuccess(int a[][COLS], int rows){
341 //Declaration the Variables
342 bool flag=false; //Set the flag
343 int sumR1=0,sumR2=0,sumR3=0; //The sum of each row
344 int sumC1=0,sumC2=0,sumC3=0; //The sum of each column
345 //Process
346 for(int j=0; j<COLS; j++){
347 sumR1+=a[0][j];
348 sumR2+=a[1][j];
349 sumR3+=a[2][j];
350 }
351 for(int i=0; i<rows; i++){
352 sumC1+=a[i][0];
353 sumC2+=a[i][1];
354 sumC3+=a[i][2];
355 }
356 //Determine the sum equal 15 or not
357 if(sumR1==15&&sumR2==15&&sumR3==15&&
358 sumC1==15&&sumC2==15&&sumC3==15) flag=true;
359 //Finish and return the flag
360 return flag;
361 }
362
192:49
INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int:
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[]
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
362
363 //00000001111111111222222222233333333334444444444555555555566666666667777777778
364 //34567890123456789012345678901234567890123456789012345678901234567890123456789
365 //**************************    isFinish    ***************************************
366 //Purpose:  Determine whether the answer is successful or not                     *
367 //Inputs:   Inputs to the function here -> Description, Range, Units              *
368 //  a[][COLS]->The integer array holding with answer                               *
369 //  rows->The rows of the array                                                   *
370 //Output:   Outputs to the function here -> Description, Range, Units             *
371 //  Return the boolean flag                                                       *
372 //*******************************************************************************
373
374 bool isFinish(int a[][COLS], int rows){
375     //Declaration the Variables
376     bool flag=false; //Set the flag
377     int times=0;     //Set the time holding the times of exist of same number
378     for(int i=0; i<rows; i++){
379         for(int j=0; j<COLS; j++){
380             if(a[i][j]==0) times++;
381         }
382     }
383     if(times!=0) flag=true;
384     //Finish and return flag
385     return flag;
386 }
387
388 //00000001111111111222222222233333333334444444444555555555566666666667777777778
389 //34567890123456789012345678901234567890123456789012345678901234567890123456789
390 //**************************   clearPosition   ***************************************
391 //Purpose:  Get the position and clear the value of this position                 *
392 //Inputs:   Inputs to the function here -> Description, Range, Units              *
393 //  a[][COLS]->The integer array holding with answer                               *
394 //  rows->The rows of the array                                                   *
```

192:49 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[]|
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
398     //  Print out the new array after clear position value                  *
399     //*******************************************************************
400
401 void clearPosition(int a[][COLS], int rows, int x1, int y1){
402         //Declaration the Variables
403         int x, y;
404         //Process
405         //Get the position want to clear
406         x=getRow(rows);
407         y=getCol(COLS);
408         //Limit the original number which can not be cleared.
409         while((x-1==x1&&y-1==y1)){
410             cout<<"The question number can not be cleared."<<endl;
411             cout<<"Please re-enter in the position."<<endl;
412             x=getRow(rows);
413             y=getCol(COLS);
414         }
415         //Check the position whether has a number
416         while(!isPositionHasNum(a,x,y)){
417             //if the position does not have a number
418             cout<<"This position ("<<x<<","<<y
419                     <<") does not have a number need to clear"<<endl;
420             cout<<"Please re-enter the position you want to clear."<<endl;
421             x=getRow(rows);
422             y=getCol(COLS);
423         }
424         //Set the position to 0
425         a[x-1][y-1]=0;
426         //Print out the new array
427         prntAry(a,rows);
428 }
429
430 //0000000111111111122222222223333333333444444444455555555556666666666777777777
```

192:49 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[][
- isSuccess(int a[][], int rov
- main(int argc, char** arg

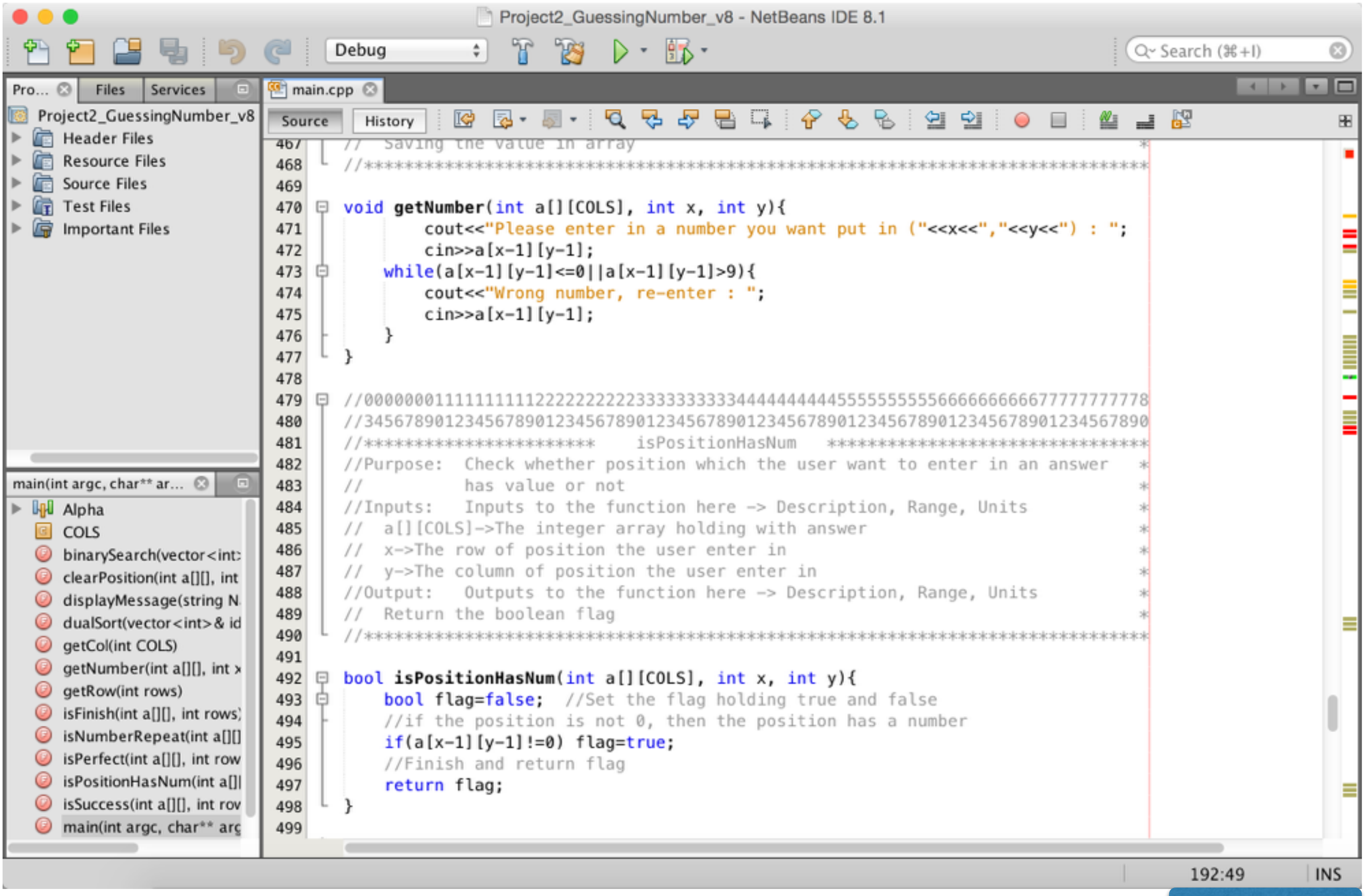
main.cpp

Source | History

```
441 //*************************************************************************
442
443 bool isNumberRepeat(int a[][COLS], int rows, int x, int y){
444     //Declaration the variables
445     int times=0; //The times one number displays
446     bool found=false; //Set the flag
447     for(int i=0; i<rows; i++){
448         for(int j=0; j<COLS; j++){
449             if(a[x-1][y-1]==a[i][j]){
450                 times++;
451             }
452         }
453     }if(times!=1) found=true;
454     //Finish and return the found
455     return found;
456 }
457
458 //0000000111111111222222222233333333334444444445555555555666666666677777777778
459 //3456789012345678901234567890123456789012345678901234567890123456789012345678890
460 //**************************    getNumber    ***************************************
461 //Purpose:  Let the user enter in a value for answer                              *
462 //Inputs:   Inputs to the function here -> Description, Range, Units              *
463 //  a[][COLS]->The integer array holding with answer                              *
464 //  x->The row of position the user enter in                                     *
465 //  y->The column of position the user enter in                                  *
466 //Output:   Outputs to the function here -> Description, Range, Units            *
467 //  Saving the value in array                                                    *
468 //*************************************************************************
469
470 void getNumber(int a[][COLS], int x, int y){
471         cout<<"Please enter in a number you want put in ("<<x<<","<<y<<") : ";
472         cin>>a[x-1][y-1];
473     while(a[x-1][y-1]<=0||a[x-1][y-1]>9){
```

192:49 | INS

Back Menu


Project2_GuessingNumber_v8 - NetBeans IDE 8.1
Debug
Search (⌘+I)
Pro...
Files
Services
Project2_GuessingNumber_v8
Header Files
Resource Files
Source Files
Test Files
Important Files
main(int argc, char** ar...
Alpha
COLS
binarySearch(vector<int>
clearPosition(int a[][], int
displayMessage(string N
dualSort(vector<int>& id
getCol(int COLS)
getNumber(int a[][], int x
getRow(int rows)
isFinish(int a[][], int rows)
isNumberRepeat(int a[][]
isPerfect(int a[][], int row
isPositionHasNum(int a[][
isSuccess(int a[][], int rov
main(int argc, char** arg
main.cpp
Source
History
// Saving the value in array
//**********
void getNumber(int a[][COLS], int x, int y){
cout<<"Please enter in a number you want put in ("<<x<<","<<y<<") : ";
cin>>a[x-1][y-1];
while(a[x-1][y-1]<=0||a[x-1][y-1]>9){
cout<<"Wrong number, re-enter : ";
cin>>a[x-1][y-1];
}
}
//0000000111111111122222222223333333333444444444455555555556666666666777777777 8
//3456789012345678901234567890123456789012345678901234567890123456789012345678 90
//************************ isPositionHasNum **********
//Purpose: Check whether position which the user want to enter in an answer *
// has value or not *
//Inputs: Inputs to the function here -> Description, Range, Units *
// a[][COLS]->The integer array holding with answer *
// x->The row of position the user enter in *
// y->The column of position the user enter in *
//Output: Outputs to the function here -> Description, Range, Units *
// Return the boolean flag *
//**********
bool isPositionHasNum(int a[][COLS], int x, int y){
bool flag=false; //Set the flag holding true and false
//if the position is not 0, then the position has a number
if(a[x-1][y-1]!=0) flag=true;
//Finish and return flag
return flag;
}
192:49
INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...
- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[][
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
510 int getCol(int COLS){
511     //Declaration the Variables
512     int col; //The number of column
513     //Input the column
514     cout<<"Please enter in the number of column : ";
515     cin>>col;
516     //check the range
517     while(col<1||col>COLS){
518         cout<<"Wrong number for column, re-enter in number (1,2,3): ";
519         cin>>col;
520     }
521     //Return the value of column
522     return col;
523 }
524
525 //0000000111111111122222222223333333333444444444455555555556666666666777777777778
526 //3456789012345678901234567890123456789012345678901234567890123456789012345678890
527 //****************************    getCol    ***************************************
528 //Purpose:  Let the user enter in the row of position for answer                  *
529 //Inputs:   Inputs to the function here -> Description, Range, Units             *
530 //  rows->The row of answer array                                                *
531 //Output:   Outputs to the function here -> Description, Range, Units            *
532 //  Return the row of position the user want to enter in a answer                *
533 //*********************************************************************************
534
535 int getRow(int rows){
536     //Declaration the Variables
537     int row; //The number of row
538     //Input the number of row
539     cout<<"Please enter in the number of row : ";
540     cin>>row;
541     //check the range
542     while(row<1||row>rows){
```

192:49 | INS

Back Menu

Project2_GuessingNumber_v8 - NetBeans IDE 8.1

Debug

Search (⌘+I)

Pro... | Files | Services

Project2_GuessingNumber_v8
- Header Files
- Resource Files
- Source Files
- Test Files
- Important Files

main(int argc, char** ar...

- Alpha
- COLS
- binarySearch(vector<int>
- clearPosition(int a[][], int
- displayMessage(string N
- dualSort(vector<int>& id
- getCol(int COLS)
- getNumber(int a[][], int x
- getRow(int rows)
- isFinish(int a[][], int rows)
- isNumberRepeat(int a[][]
- isPerfect(int a[][], int row
- isPositionHasNum(int a[][
- isSuccess(int a[][], int rov
- main(int argc, char** arg

main.cpp

Source | History

```
550 //0000000111111111122222222223333333333444444444455555555556666666667777777778
551 //34567890123456789012345678901234567890123456789012345678901234567890123456789
552 //**************************    prntAry    ***************************************
553 //Purpose:  Display the answer array                                             *
554 //Inputs:   Inputs to the function here -> Description, Range, Units            *
555 //  a[][COLS]->The integer array holding with answer                             *
556 //  row->The row of answer array                                                *
557 //Output:   Outputs to the function here -> Description, Range, Units           *
558 //  Print out the answer array                                                  *
559 //*******************************************************************************
560
561 void prntAry(int a[][COLS], int row){
562     //display the array
563     cout<<"┌─┬─┬─┐"<<endl;
564     for(int i=0; i<row; i++){
565         if(i!=0) cout<<"├─┼─┼─┤"<<endl;
566         for(int j=0; j<COLS; j++){
567             if(a[i][j]==0) cout<<"|  ";
568             else cout<<"| "<<a[i][j]<<" ";
569             if(j%3==2) cout<<"|"<<endl;
570         }
571     }cout<<"└─┴─┴─┘"<<endl;
572     cout<<endl;
573 }
574
575
576
577
```

192:49 | INS

Back Menu