# Project 1

<Battleship>

Manjot Dhindsa

07/21/15

# Introduction

## Title: Battleship Game

The game battleship is played with two players. Each player has five ships that they place on the board without telling the other player their ships location. The game starts with each player choosing the coordinates that they want to place their ships at and after the first player has done this then the second player does so as well. The players then take turns trying to sink the opponents' ships.

# Summary:

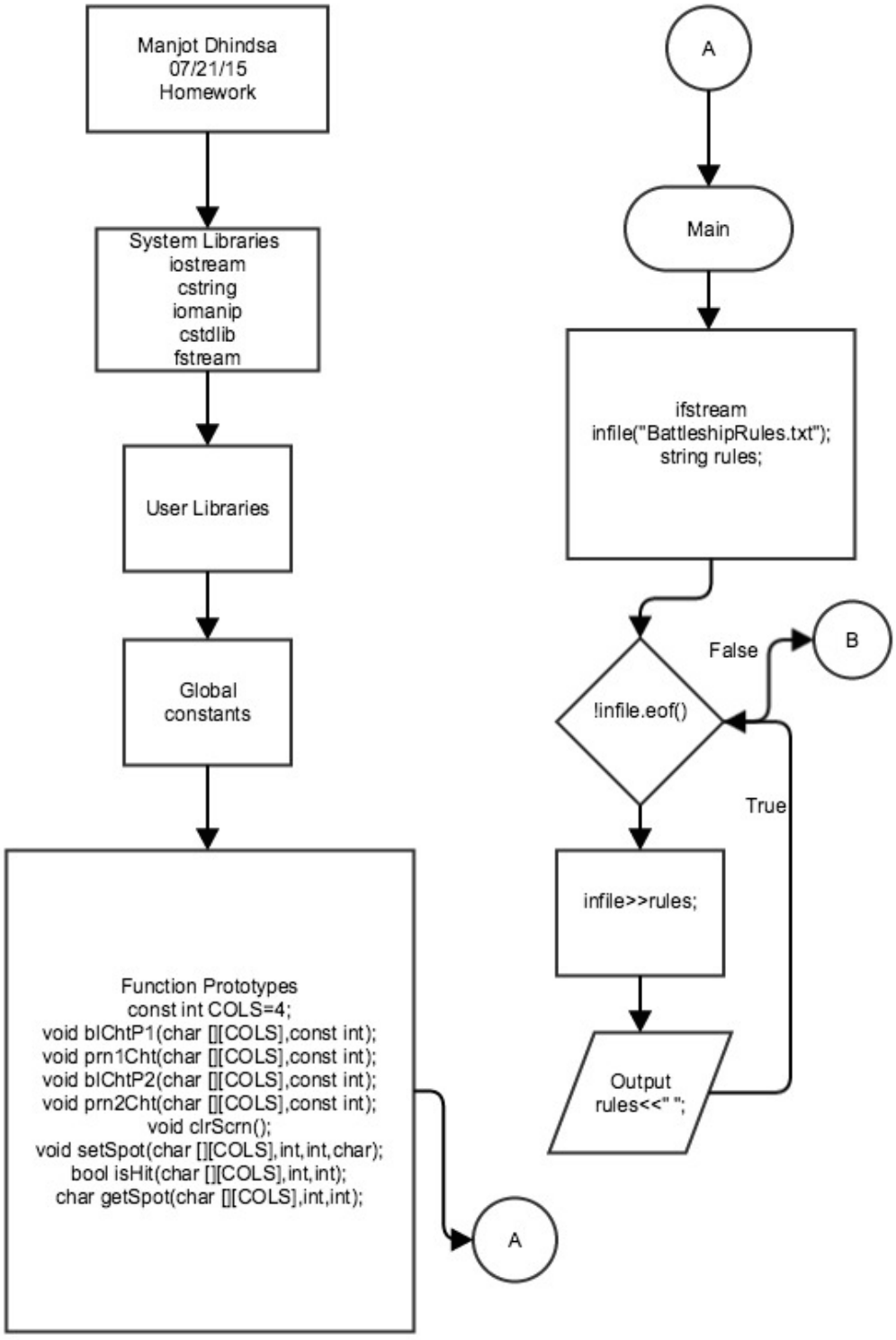
Project size: 291 lines
The number of variables: 11
The number of functions: 8

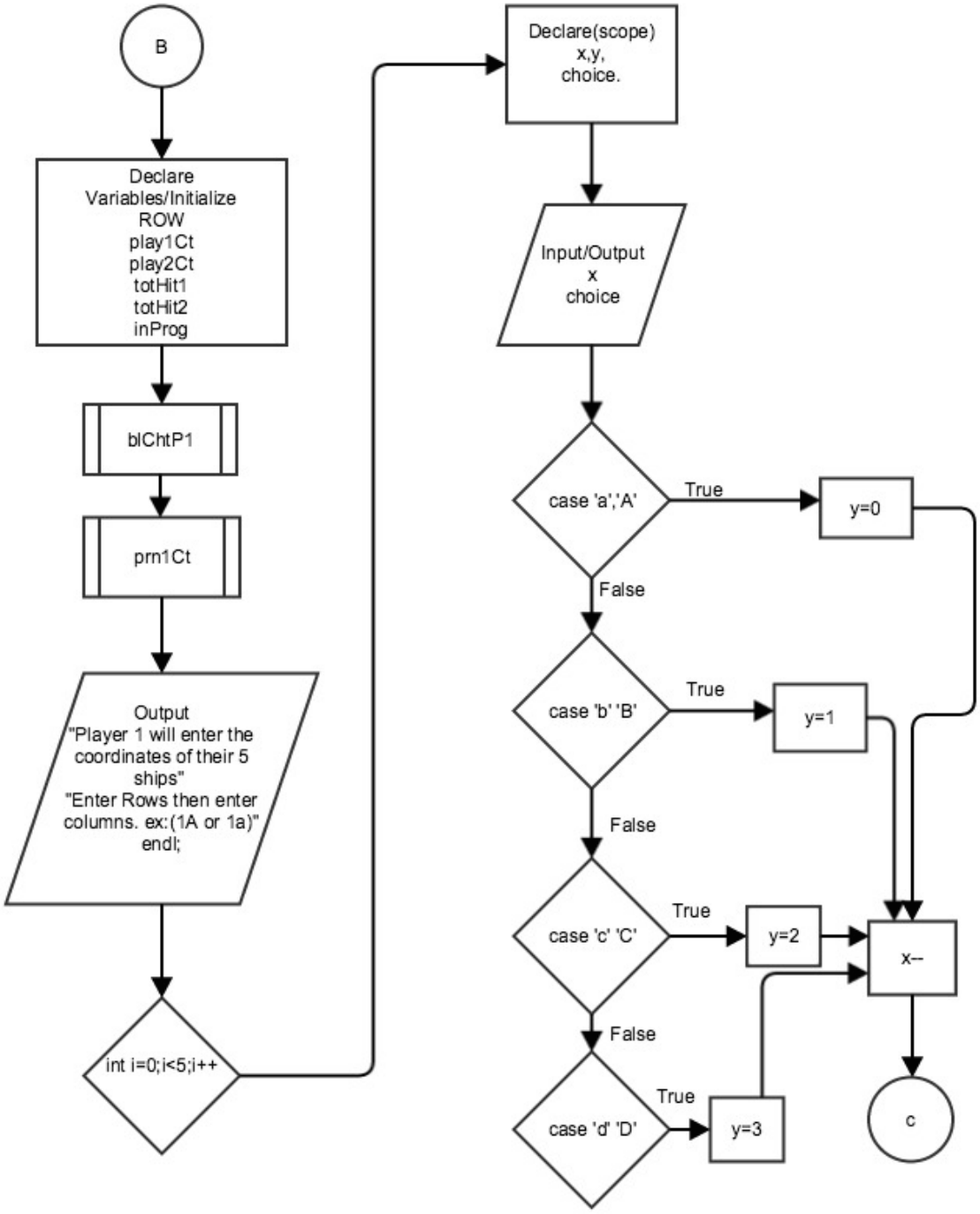
It took me about a week to complete this project and in that time I was able to use 2 dimensional arrays to make my grid for each player. It was hard making it so that the player couldn't hit the same spot multiple times and still get a hit but with some help I was able to fix that problem. Even though there was more I was hoping I could do with this game I just about ran out of time so I'm going to add those changes in for the Project 2. What was probably the hardest part of this project was fixing the parts where it would allow the user to hit the same spot over and over again and still consider it a hit. Once again with help I was able to understand what I had to do to fix that problem. Overall this first project was a great learning experience and I learned a lot about arrays while making this game.

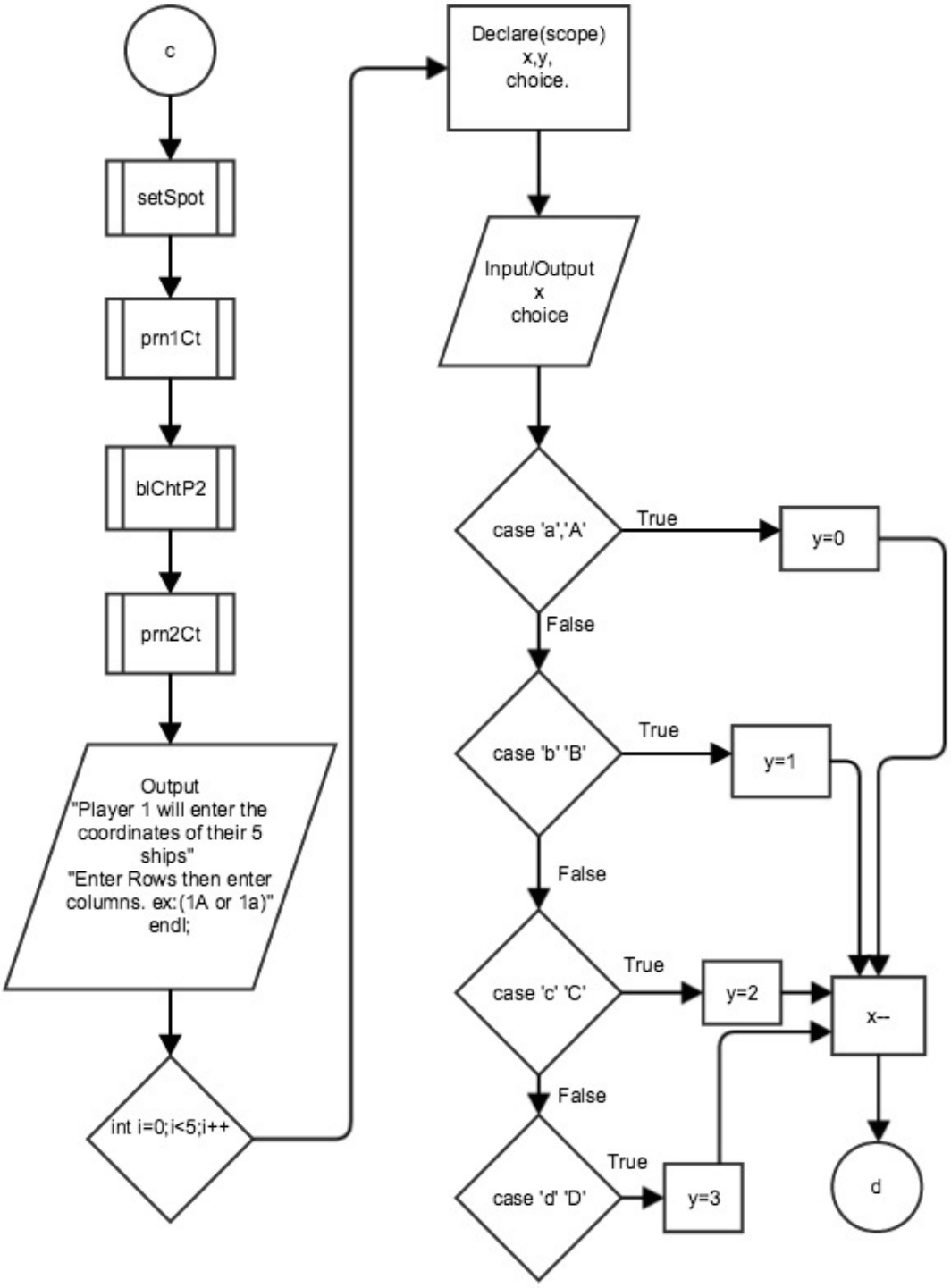
# Description:

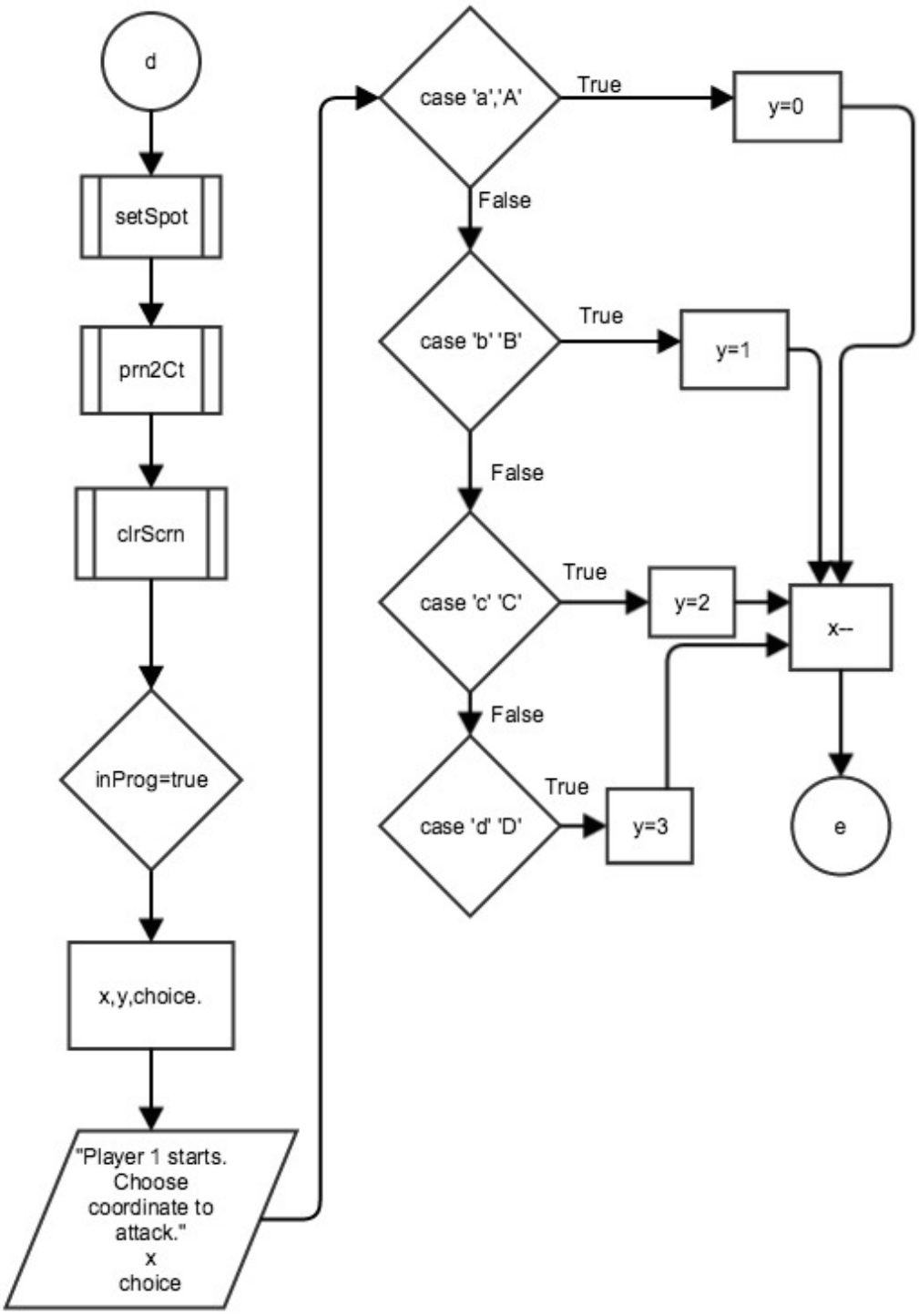
The user enters coordinates to attack the other player. I used a function to make it that after that position is hit if the user tries to hit it again it will print out that its already been hit.

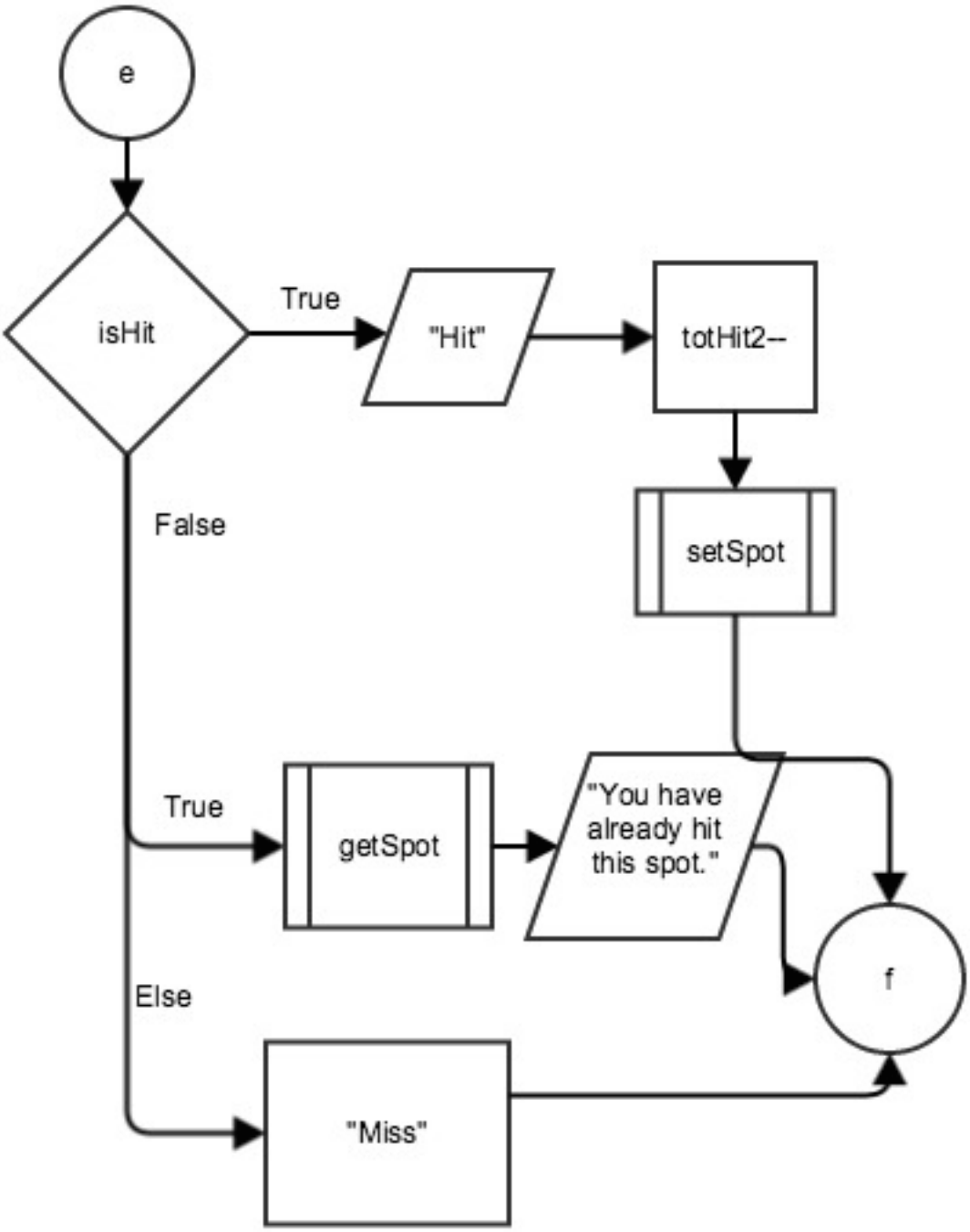
# Flow Chart:

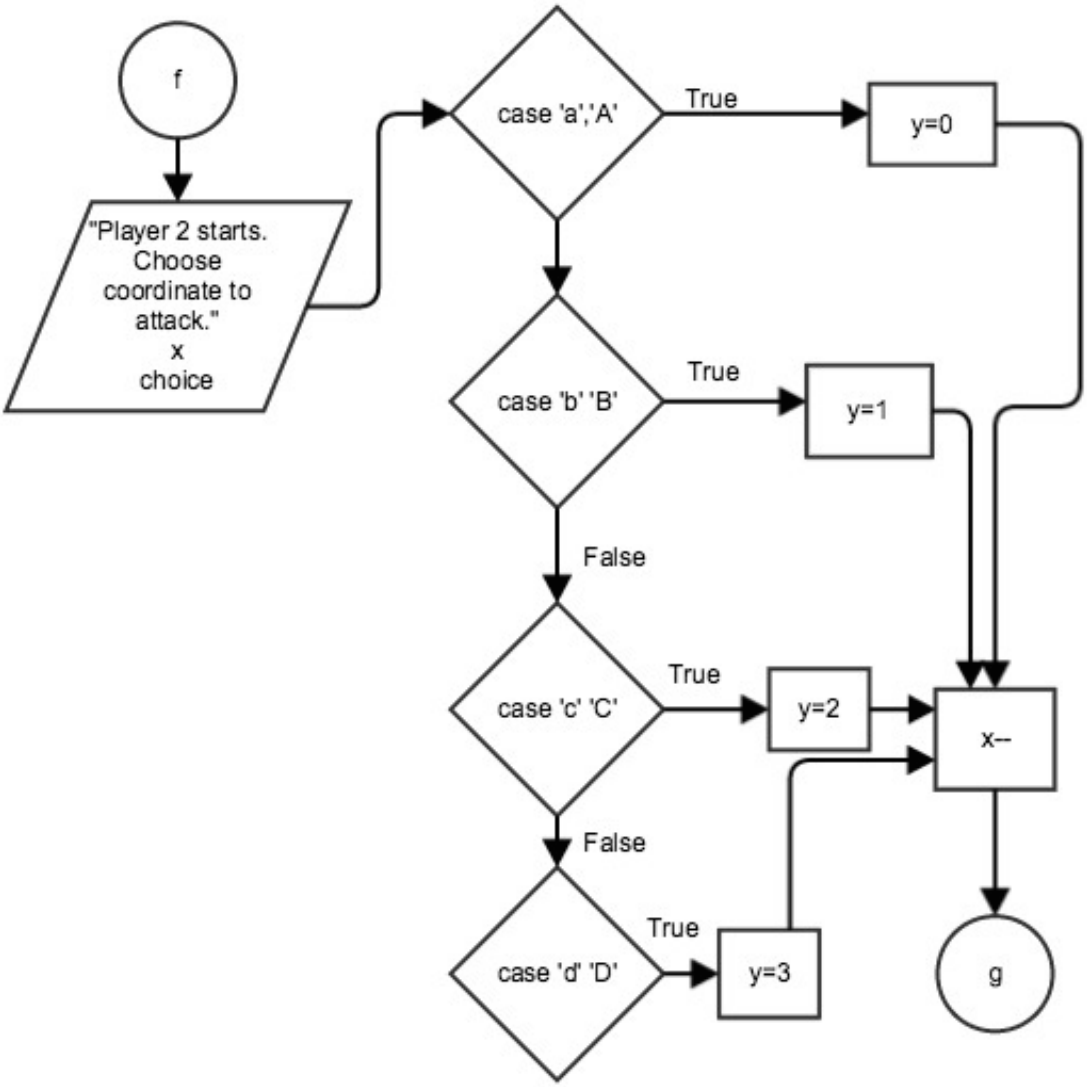

Manjot Dhindsa
07/21/15
Homework
System Libraries
iostream
cstring
iomanip
cstdlib
fstream
User Libraries
Global
constants
Function Prototypes
const int COLS=4;
void blChtP1(char [][COLS],const int);
void prn1Cht(char [][COLS],const int);
void blChtP2(char [][COLS],const int);
void prn2Cht(char [][COLS],const int);
void clrScrn();
void setSpot(char [][COLS],int,int,char);
bool isHit(char [][COLS],int,int);
char getSpot(char [][COLS],int,int);
A
A
Main
ifstream
infile("BattleshipRules.txt");
string rules;
!infile.eof()
False
B
True
infile>>rules;
Output
rules<<" ";

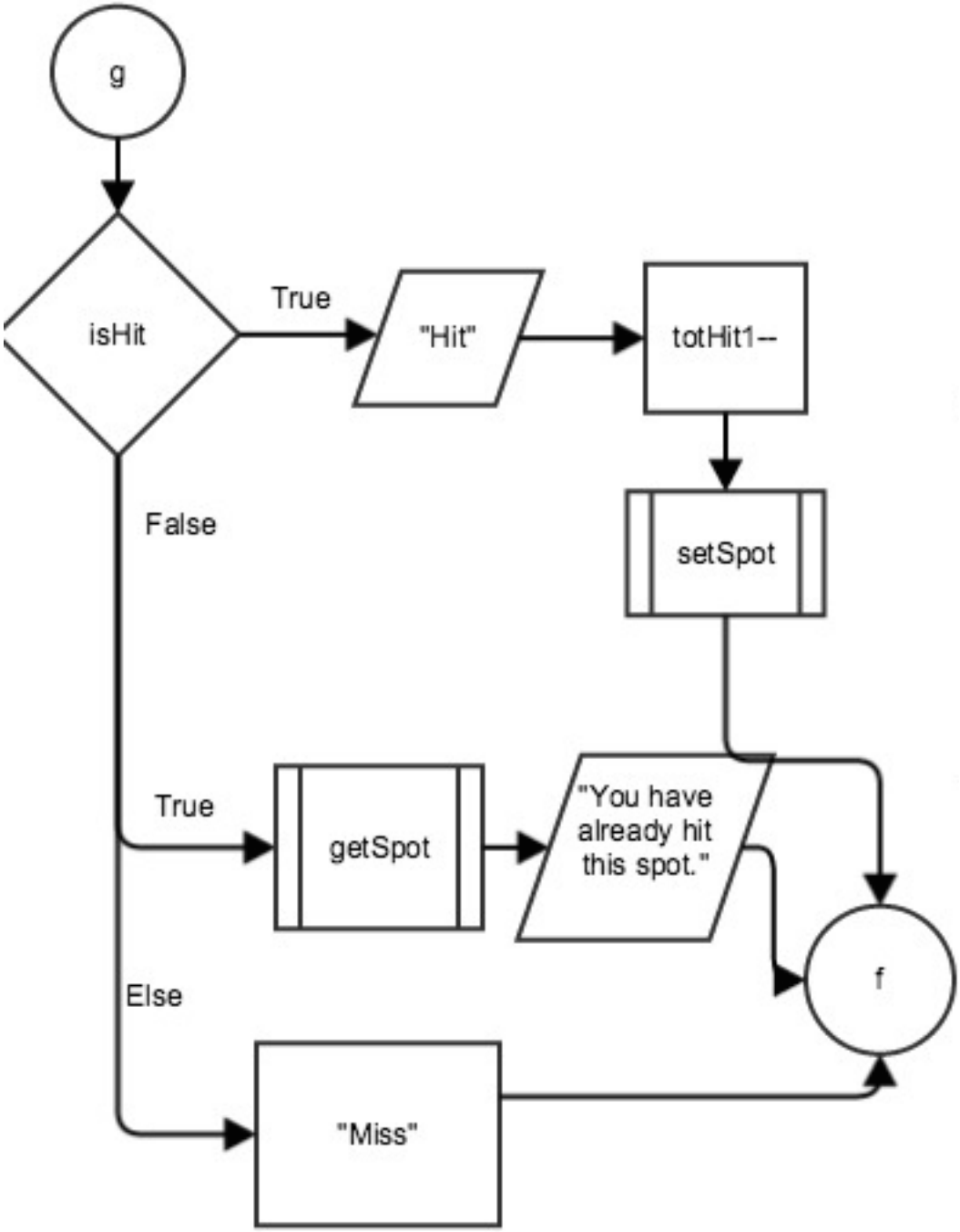

B
Declare
Variables/Initialize
ROW
play1Ct
play2Ct
totHit1
totHit2
inProg
blChtP1
prn1Ct
Output
"Player 1 will enter the coordinates of their 5 ships"
"Enter Rows then enter columns. ex:(1A or 1a)"
endl;
int i=0;i<5;i++
Declare(scope)
x,y,
choice.
Input/Output
x
choice
case 'a','A'
True
y=0
False
case 'b' 'B'
True
y=1
False
case 'c' 'C'
True
y=2
False
case 'd' 'D'
True
y=3
x--
c

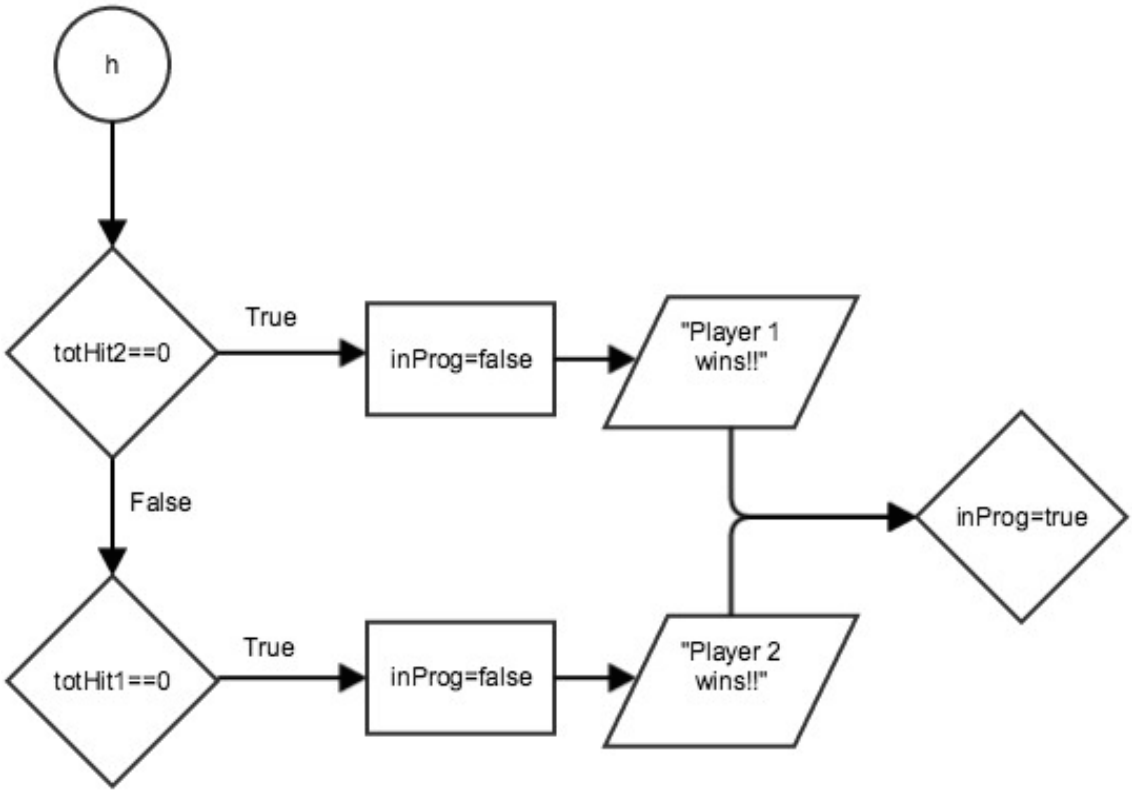

c
setSpot
prn1Ct
blChtP2
prn2Ct
Output
"Player 1 will enter the coordinates of their 5 ships"
"Enter Rows then enter columns. ex:(1A or 1a)"
endl;
int i=0;i<5;i++
Declare(scope)
x,y,
choice.
Input/Output
x
choice
case 'a','A'
True
y=0
False
case 'b' 'B'
True
y=1
False
case 'c' 'C'
True
y=2
x--
False
case 'd' 'D'
True
y=3
d

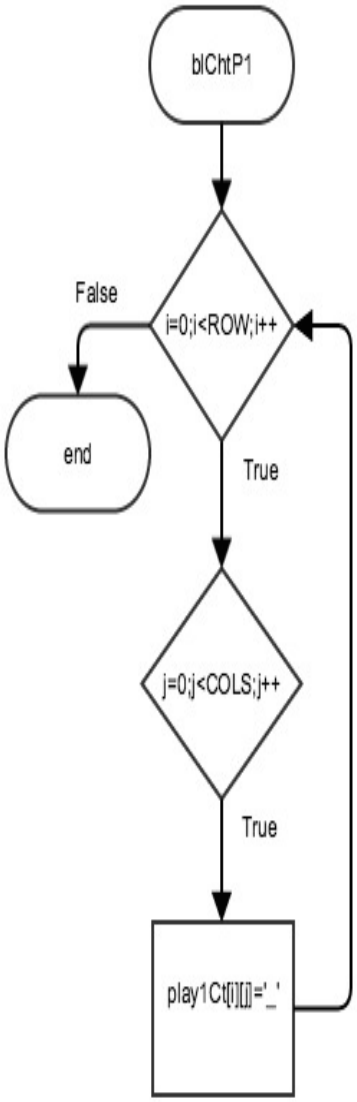

d
setSpot
prn2Ct
clrScrn
inProg=true
x,y,choice.
"Player 1 starts.
Choose
coordinate to
attack."
x
choice
case 'a','A'
True
y=0
False
case 'b' 'B'
True
y=1
False
case 'c' 'C'
True
y=2
False
case 'd' 'D'
True
y=3
x--
e

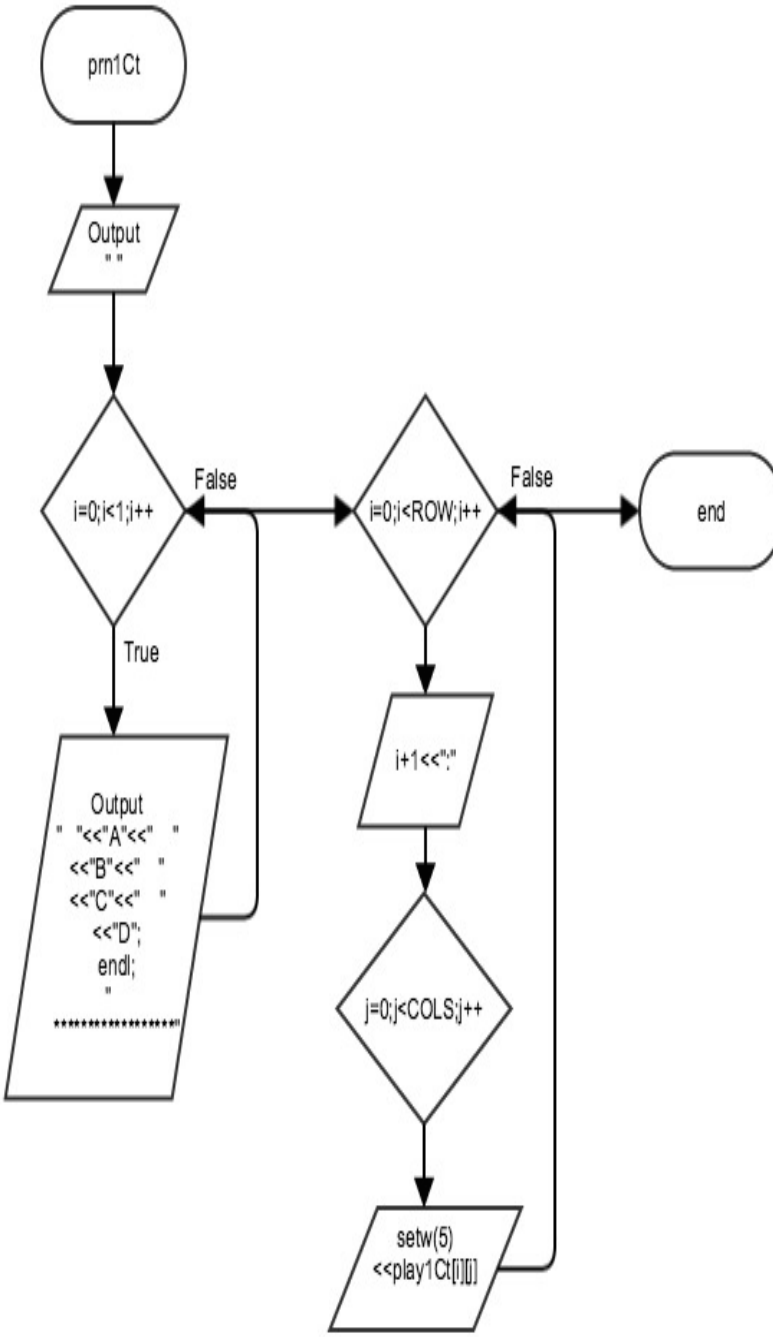

e
isHit
True
"Hit"
totHit2--
setSpot
False
True
getSpot
"You have already hit this spot."
f
Else
"Miss"

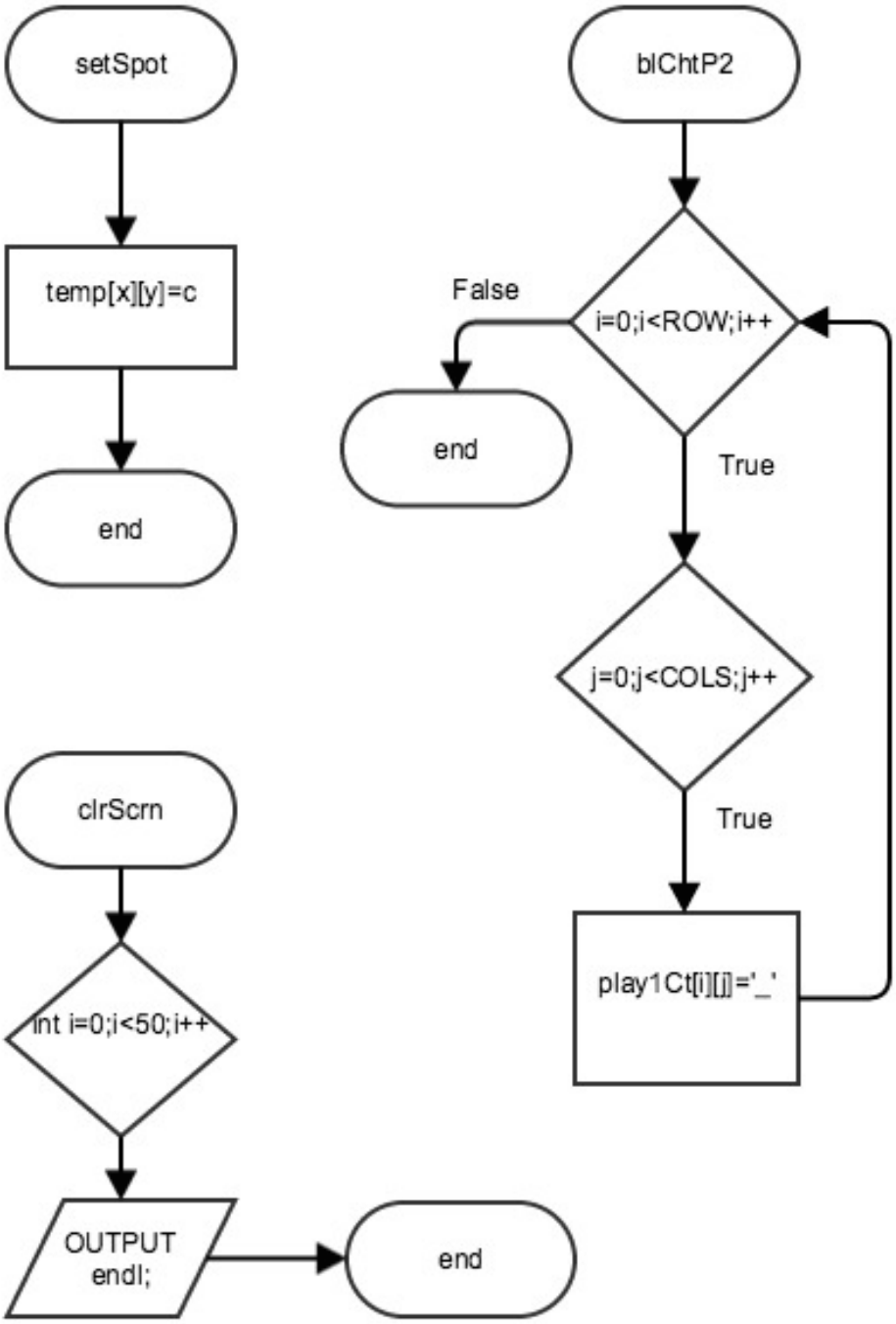

f
"Player 2 starts.
Choose
coordinate to
attack."
x
choice
case 'a','A'
True
y=0
case 'b' 'B'
True
y=1
False
case 'c' 'C'
True
y=2
x--
False
case 'd' 'D'
True
y=3
g

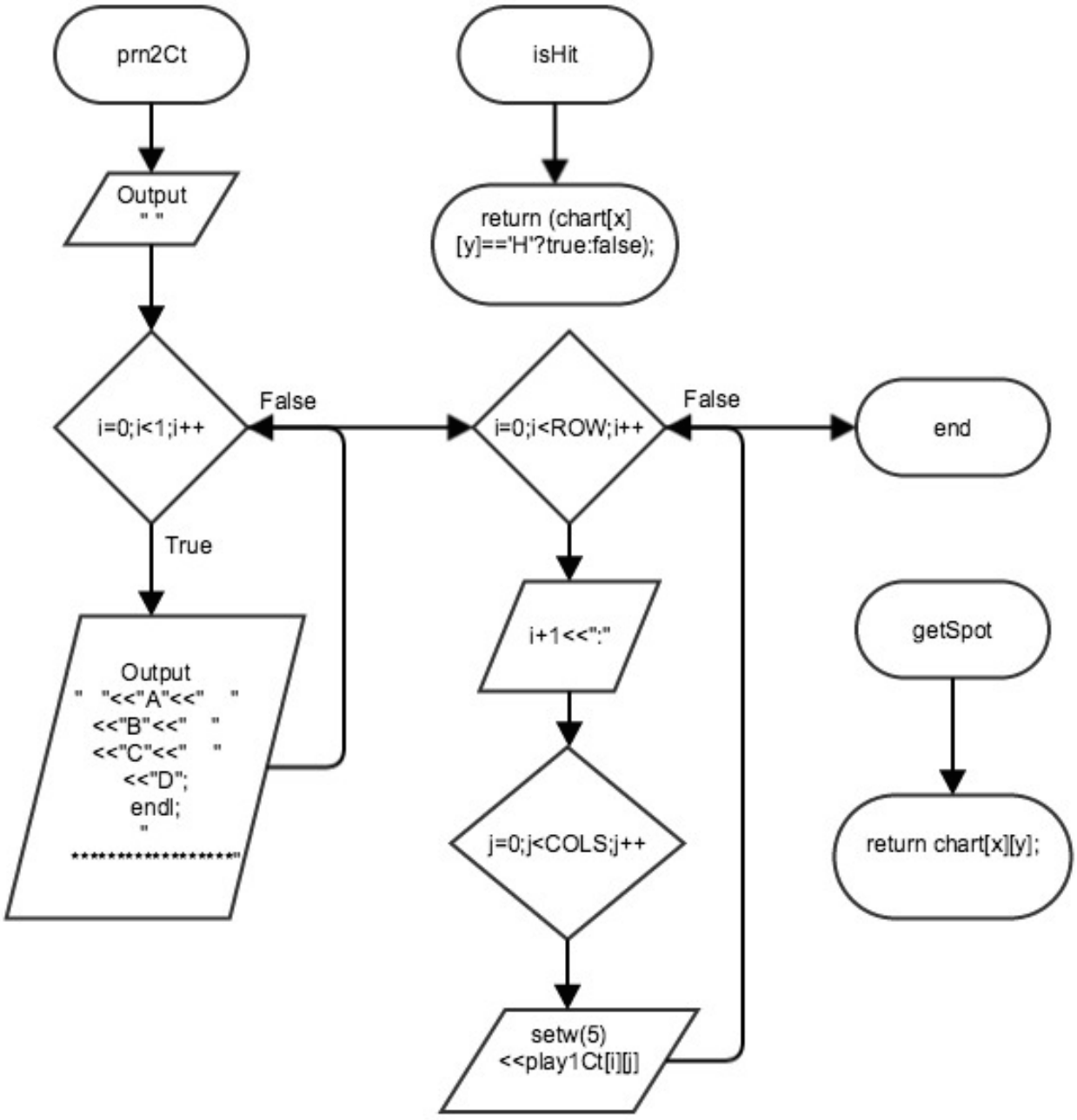

g
isHit
True
"Hit"
totHit1--
setSpot
False
True
getSpot
"You have already hit this spot."
f
Else
"Miss"


h
totHit2==0
True
inProg=false
"Player 1 wins!!"
False
totHit1==0
True
inProg=false
"Player 2 wins!!"
inProg=true


blChtP1
i=0;i<ROW;i++
False
end
True
j=0;j<COLS;j++
True
play1Ct[i][j]='_'


prn1Ct
Output
" "
i=0;i<1;j++
False
True
Output
" "<<"A"<<" "
<<"B"<<" "
<<"C"<<" "
<<"D";
endl;
"
**********"
i=0;i<ROW;i++
False
end
i+1<<":"
j=0;j<COLS;j++
setw(5)
<<play1Ct[i][j]


setSpot
temp[x][y]=c
end
blChtP2
i=0;i<ROW;i++
False
end
True
j=0;j<COLS;j++
True
play1Ct[i][j]='_'
clrScrn
int i=0;i<50;i++
OUTPUT
endl;
end


prn2Ct
Output
" "
i=0;i<1;i++
False
True
Output
" "<<"A"<<" "
<<"B"<<" "
<<"C"<<" "
<<"D";
endl;
"
**********"
isHit
return (chart[x]
[y]=='H'?true:false);
i=0;i<ROW;i++
False
end
i+1<<":"
j=0;j<COLS;j++
setw(5)
<<play1Ct[i][j]
getSpot
return chart[x][y];

# Pseudo code:

Bring in file for the description of the game.
File has the description of the game and how to play.

Declare variables for battleship chart.
ROW=4 is the amount of rows for the table.
play1Ct[ROW][COLS]  this is the table for player 1.
play2Ct[ROW][COLS]  this is the table for player 2.
totHit1=5 this will decrement after each hit by player 2.
totHit2=5 this will decrement after each hit by player 1.
inProg=true keeps the do while loop working.

Call BattleShip chart function Player 1.
For loop for the first player to choose where they would like to position their ships. Int x,y, char choice are locally defined in this scope for player one.
Player inputs : x and choice.
Then switch statement for choice. If
choice = a,A then y=0.
If choice=b,B then y=1.
If choice = c,C then y=2.
If choice = d,D then y=3.
X decrements as this loop happens so if player 1 enters 1 for x it takes it as a 0. The setSpot function is then called after each coordinate is placed and sets that spot to a 'H'. The loop then ends and it is then player 2 turn.

Call BattleShip chart function Player 2.
Same as above happens but for player 2.

A do while loop for the actual game.
do{
Initialize x,y, and choice for the do while loop
Ask player one to input x and choice
Switch (choice)
    Case a,A: y=0
    Case b,B: y=1
    Case c,C: y=2
    Case d,D y=3
Then x decrements (x--)

Also the isHit function gets called
If isHit: outputs “Hit” and the totHit2-- total ships for player 2 decrements.
The setSpot function is also called to make the hit spot change to an X so that it can’t be hit again.
Else If getSpot: output “You have already hit this spot.” So that after the setSpot function changes the spot to an X so that if you try to hit the spot again it outputs that.
Else outputs “Miss”

This process happens again for player 2.

Then:
If totHit2==0: if the player 1 hits all of player 2 ships then the loop ends and player 1 wins. The program outputs: “Player 1 wins!!” and inProg becomes false.
Else totHit1==0: if player 2 hits all of player 1 ships then the loop ends and player 2 wins. The program outputs: “Player 2 wins!!” and inProg becomes false.
}while(inProg)

## Major Variables:

| Type | Variable Name | Description | Location |
|---|---|---|---|
| Const int | ROW=4<br>COLS=4 | located in arrays to make tables | play1Ct[][COLS]<br>play2Ct[][COLS]<br>blChtP1[][COLS]<br>blChtP2[][COLS]<br>prn1Cht[][COLS]<br>prn2Cht[][COLS]<br>setSpot[][COLS]<br>getSpot[][COLS]<br>isHit[][COLS] |
| char | play1Ct | Chart for P1 | all ^ except<br>blChtP2[][COLS]<br>prn2Cht[][COLS] |
| | play2Ct | Chart for P2 | all ^ except<br>blChtP1[][COLS]<br>prn1Cht[][COLS] |
| | totHit1=5 | amount of ships | lines 163, 174 |
| | totHit2=5 | amount of ships | lines 137, 171 |
| | choice | A,B,C,D | lines 62,64,65,<br>94,96,97,120,<br>122,123,148,149 |
| | c | used in function to change – to ‘H’ | setSpot function |
| bool | inProg | for do-while loop | lines 172,175,178 |
| int | x,y | coordinates | every time player inputs their coordinates |

# Program

```
/*
 * File:   main.cpp
 * Author: Manjot Dhindsa
 * Created on July 15, 2015, 9:02 PM
 * Purpose: BattleShip Game
 */

//System Libraries
#include <iostream>
#include <cstring>
#include <iomanip>
#include <cstdlib>
#include <fstream>
using namespace std; //std namespace -> iostream

//User Libraries

//Global constants

//Function Prototypes
const int COLS=4;
void blChtP1(char [][COLS],const int);
void prn1Cht(char [][COLS],const int);
void blChtP2(char [][COLS],const int);
void prn2Cht(char [][COLS],const int);
void clrScrn();
void setSpot(char [][COLS],int,int,char);
bool isHit(char [][COLS],int,int);
char getSpot(char [][COLS],int,int);

//Execution Begins Here!
int main(int argc, char** argv) {
    //Bring in file for the description of the game.
    ifstream infile("BattleshipRules.txt");
    string   rules;
    while(!infile.eof()){
        infile>>rules;
        cout<<rules<<" ";
```

```
}

//Declare variables for battleship chart
const int ROW=4;              //Amount of rows
char      play1Ct[ROW][COLS];  //Rows by Columns Player 1
char      play2Ct[ROW][COLS];  //Rows by Columns Player 2
char      totHit1=5;          //=5 ships hits
char      totHit2=5;          //=5 ships hit
bool      inProg=true;         //means that the game's still in progress

//Call BattleShip chart function Player 1
blChtP1(play1Ct,ROW);
prn1Cht(play1Ct,ROW);
cout<<endl;

//Where does player 1 want their ships?
cout<<"Player 1 will enter the coordinates of their 5 ships"<<endl;
cout<<"Enter Rows then enter columns. ex:(1A or 1a)"<<endl;
cout<<endl;

//Loop For player 1 to decide his coordinates
for(int i=0;i<5;i++){
   int x,y;
   char choice;
   cin>>x;
   cin>>choice;
   switch(choice){
      case 'A':
      case 'a': y=0;break;
      case 'B':
      case 'b': y=1;break;
      case 'C':
      case 'c': y=2;break;
      case 'D':
      case 'd': y=3;break;
   }
   x--;
   setSpot(play1Ct,x,y,'H');
}
prn1Cht(play1Ct,ROW);
```

```
cout<<"These are the positions of your 5 ships."<<endl;

//Call BattleShip Chart Player 2
blChtP2(play2Ct,ROW);
prn2Cht(play2Ct,ROW);

//Where does player 2 want their ships?
cout<<"Player 2 will enter the coordinates of their 5 ships"<<endl;
cout<<"Enter Rows then enter columns. ex:(1A or 1a)"<<endl;
cout<<endl;

//Loop For player 2 to decide his coordinates
int i=0;
while(i<5){
   int x,y;
   char choice;
   cin>>x;
   cin>>choice;
   switch(choice){
      case 'A':
      case 'a': y=0;break;
      case 'B':
      case 'b': y=1;break;
      case 'C':
      case 'c': y=2;break;
      case 'D':
      case 'd': y=3;break;
   }
   x--;
   setSpot(play2Ct,x,y,'H');
   i++;
}
prn2Cht(play2Ct,ROW);
cout<<"These are the positions of your 5 ships."<<endl;
clrScrn();

//Actual game. Players start attacking each other
do{
   //Player 1 attacks Player 2
   cout<<"Player 1 starts. Choose coordinate to attack."<<endl;
```

```
int x,y;
char choice;
cin>>x;
cin>>choice;
switch(choice){
    case 'A':
    case 'a': y=0;break;
    case 'B':
    case 'b': y=1;break;
    case 'C':
    case 'c': y=2;break;
    case 'D':
    case 'd': y=3;break;
}
x--;
//if/else for if the player hits or misses
if(isHit(play2Ct,x,y)){
    cout<<"Hit."<<endl;
    totHit2--;
    setSpot(play2Ct,x,y,'X');
}else if(getSpot(play2Ct,x,y)=='X'){
        cout<<"You have already hit this spot."<<endl;
}else{
    cout<<"Miss."<<endl;
}

//Player 2 attacks Player 1
cout<<"Player 2 starts. Choose coordinate to attack."<<endl;
cin>>x;
cin>>choice;
switch(choice){
    case 'A':
    case 'a': y=0;break;
    case 'B':
    case 'b': y=1;break;
    case 'C':
    case 'c': y=2;break;
    case 'D':
    case 'd': y=3;break;
}
```

```
        x--;
        //if/else for if the player hits or misses
        if(isHit(play1Ct,x,y)){
            cout<<"Hit."<<endl;
            totHit1--;
            setSpot(play1Ct,x,y,'X');
        }else if(getSpot(play1Ct,x,y)=='X'){
            cout<<"You have already hit this spot"<<endl;
        }else{
            cout<<"Miss."<<endl;
        }

        if(totHit2==0){
            inProg=false;
            cout<<"Player 1 wins!!"<<endl;
        }else if(totHit1==0){
            inProg=false;
            cout<<"Player 2 wins!!"<<endl;
        }
    }while(inProg);

    //Exit Stage Right!
    return 0;
}

/*************************************************************
 ************ Makes Battleship Chart for P1 ****************
 *************************************************************
 * Purpose: Makes 4 by 4 chart
 */
void blChtP1(char play1Ct[][COLS],const int ROW){
    //Loop To Find Chart
    for(int i=0;i<ROW;i++){
        for(int j=0;j<COLS;j++){
            play1Ct[i][j]='_';
        }
        cout<<endl;
    }
}
```

```
/*************************************************************
 ********** Prints Battleship Chart P1 **********
 *************************************************************
 * Purpose: Prints out already made chart
 */
void prn1Cht(char play1Ct[][COLS],const int ROW){
   cout<<"   ";
   for(int i=1;i<=1;i++){
      cout<<"   "<<"A"<<"    "<<"B"<<"    "<<"C"<<"    "<<"D";
   }cout<<endl;
   cout<<"     ******************"<<endl;
   for(int i=0;i<ROW;i++){
      cout<<i+1<<":";
      for(int j=0;j<COLS;j++){
         cout<<setw(5)<<play1Ct[i][j];
      }
      cout<<endl;
   }
}

/*************************************************************
 ********** Makes Battleship Chart for P2 **********
 *************************************************************
 * Purpose: Makes 4 by 4 chart
 */
void blChtP2(char play2Ct[][COLS],const int ROW){
   //Loop To Find Chart
   for(int i=0;i<ROW;i++){
      for(int j=0;j<COLS;j++){
         play2Ct[i][j]='_';
      }
      cout<<endl;
   }
}

/*************************************************************
 ********** Prints Battleship Chart P2 **********
 *************************************************************
 * Purpose: Prints out already made chart
 */
```

```
void prn2Cht(char play2Ct[][COLS],const int ROW){
   cout<<"   ";
   for(int i=1;i<=1;i++){
      cout<<"   "<<"A"<<"    "<<"B"<<"    "<<"C"<<"    "<<"D";
   }cout<<endl;
   cout<<"    *****************"<<endl;
   for(int i=0;i<ROW;i++){
      cout<<i+1<<":";
      for(int j=0;j<COLS;j++){
         cout<<setw(5)<<play2Ct[i][j];
      }
      cout<<endl;
   }
}

/*************************************************************
 ********************* clear screen ************************
 *************************************************************
 * Purpose: ends fifty lines after coordinates are chosen
 */
void clrScrn(){
   for(int i=0;i<50;i++){
      cout<<endl;
   }
}

/*************************************************************
 ************************ setSpot ***************************
 *************************************************************
 * Purpose: changes spot on chart to H or X
 */
void setSpot(char temp[][COLS],int x,int y,char c){
   temp[x][y]=c;
}

/*************************************************************
 ************************ isHit *****************************
 *************************************************************
 * Purpose: checks to see if coordinate = H on chart
 */
```

```
bool isHit(char chart[][COLS],int x,int y){
   //Using Ternary Operator
   return (chart[x][y]=='H'?true:false);
}

/*************************************************************
 ************************ getSpot ************************
 *************************************************************
 * Purpose: makes it so you cant attack same spot twice
 */
char getSpot(char chart[][COLS],int x,int y){
   return chart[x][y];
}
```