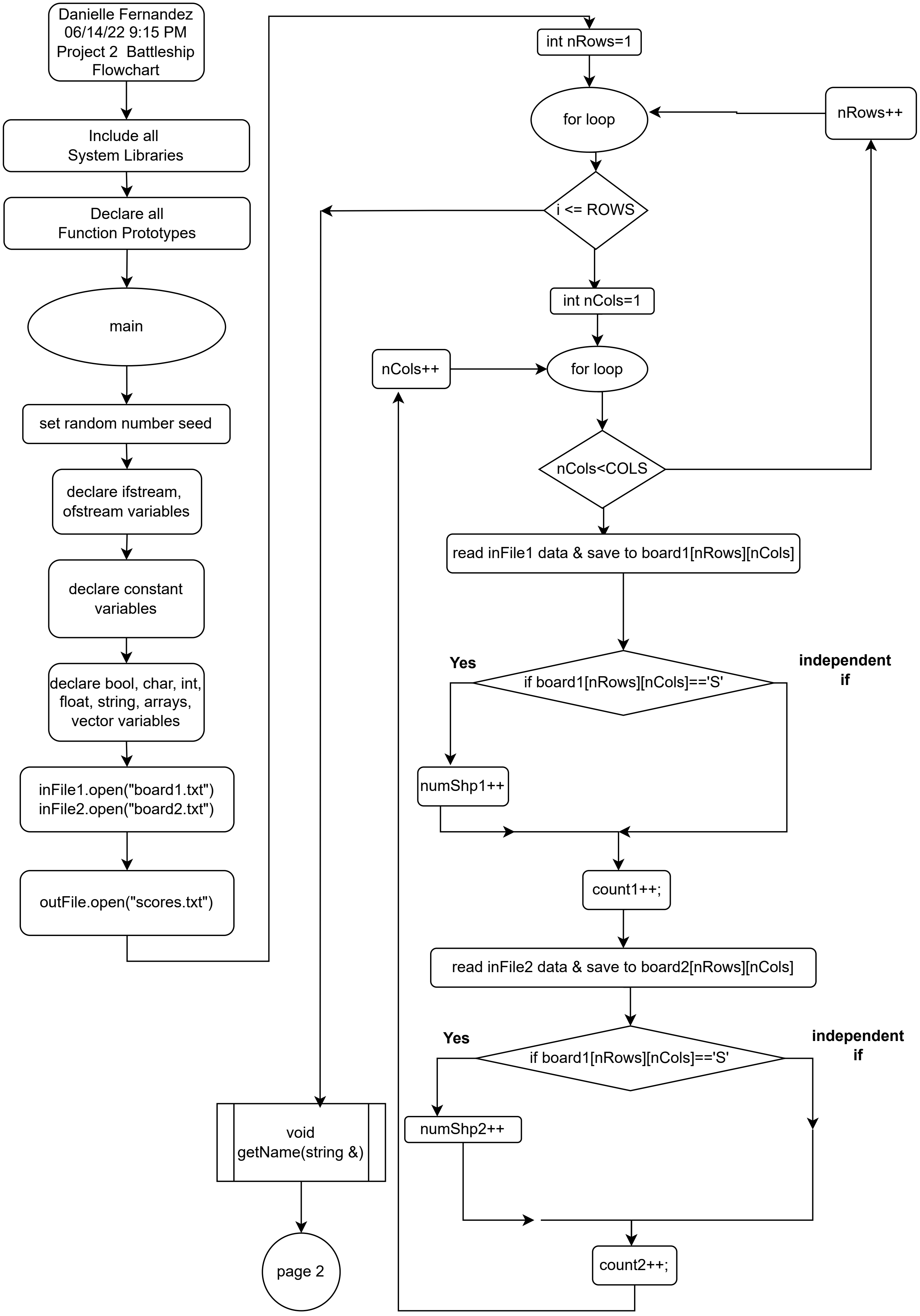
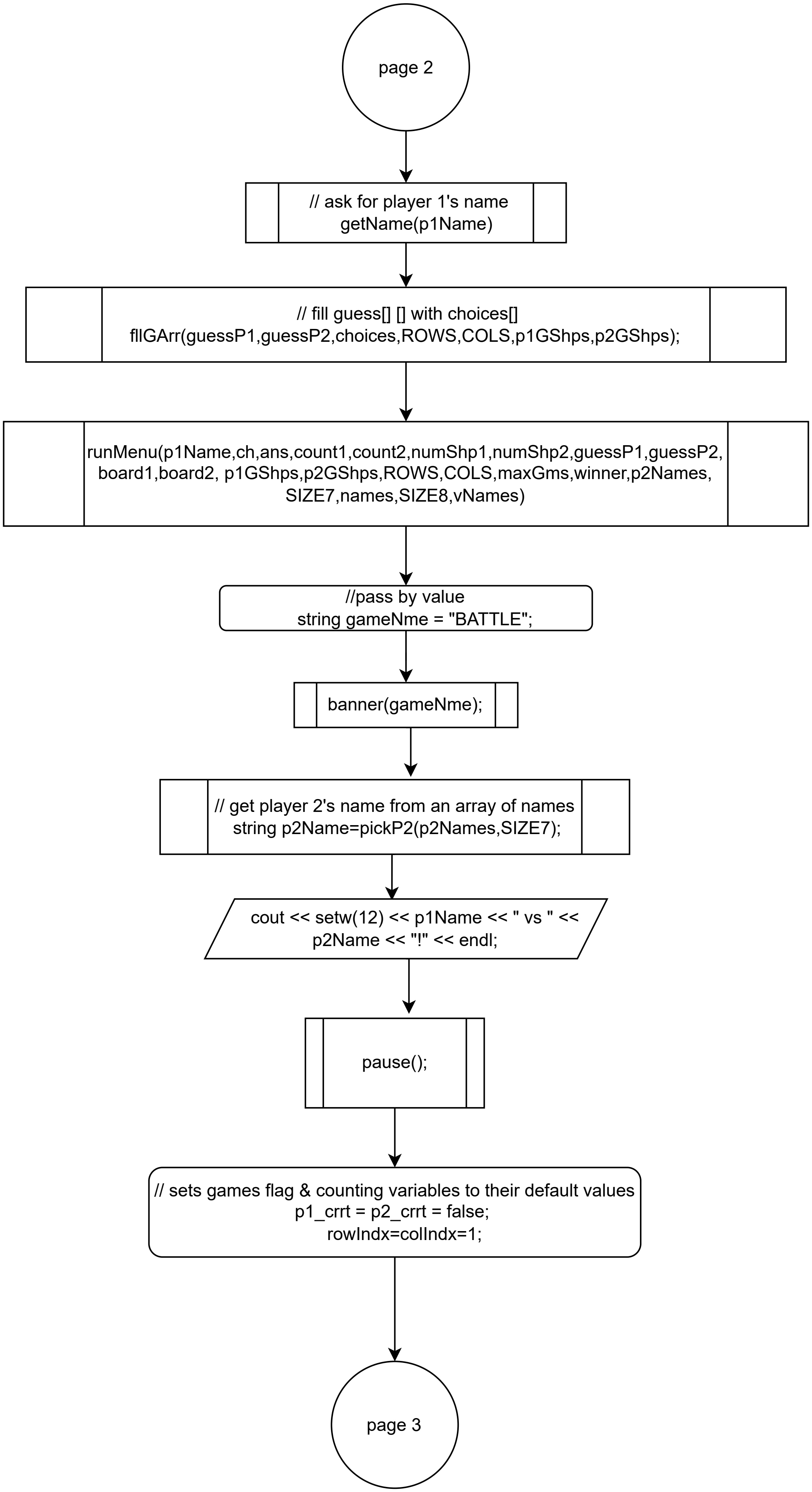
Battleship FlowChart

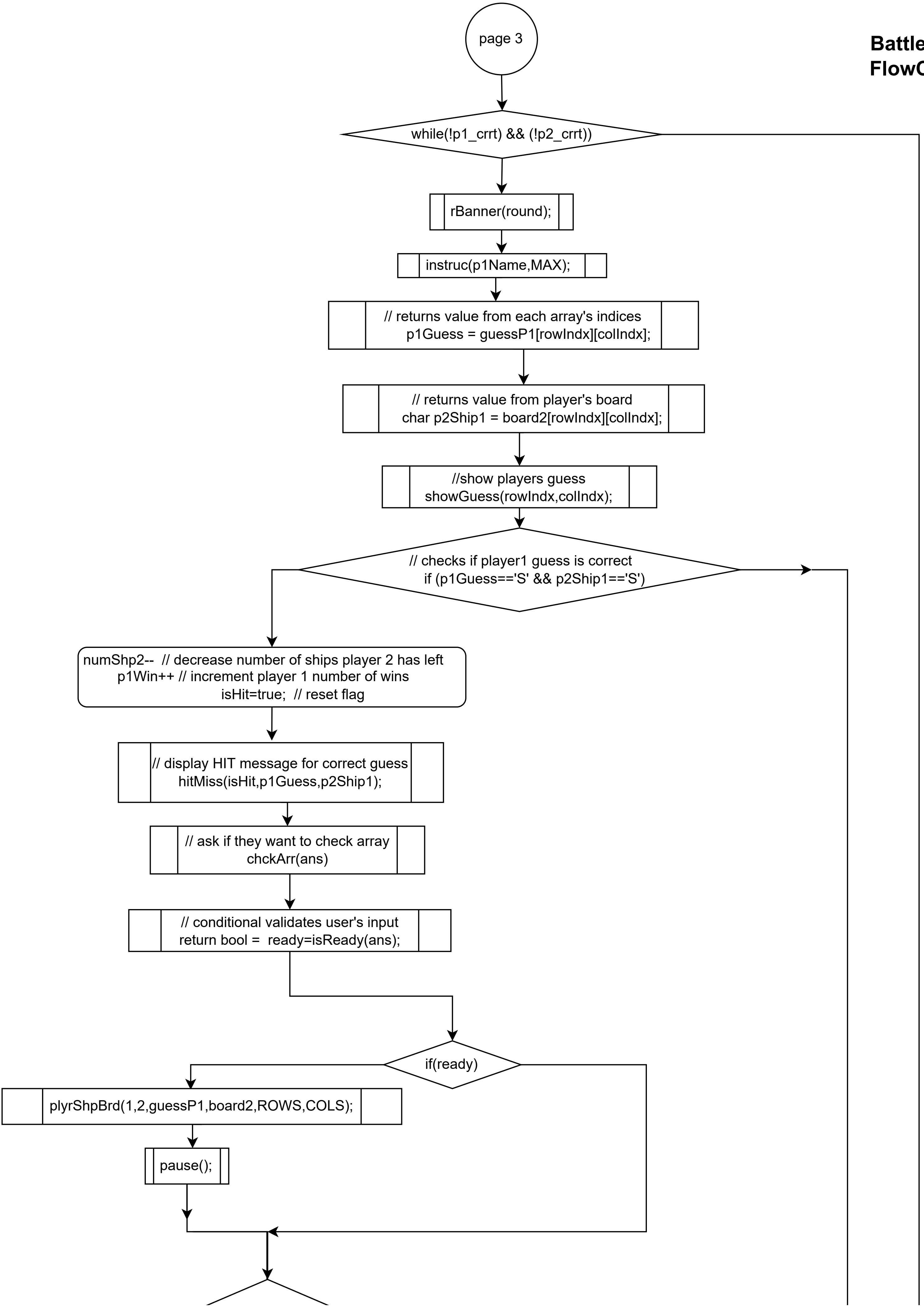


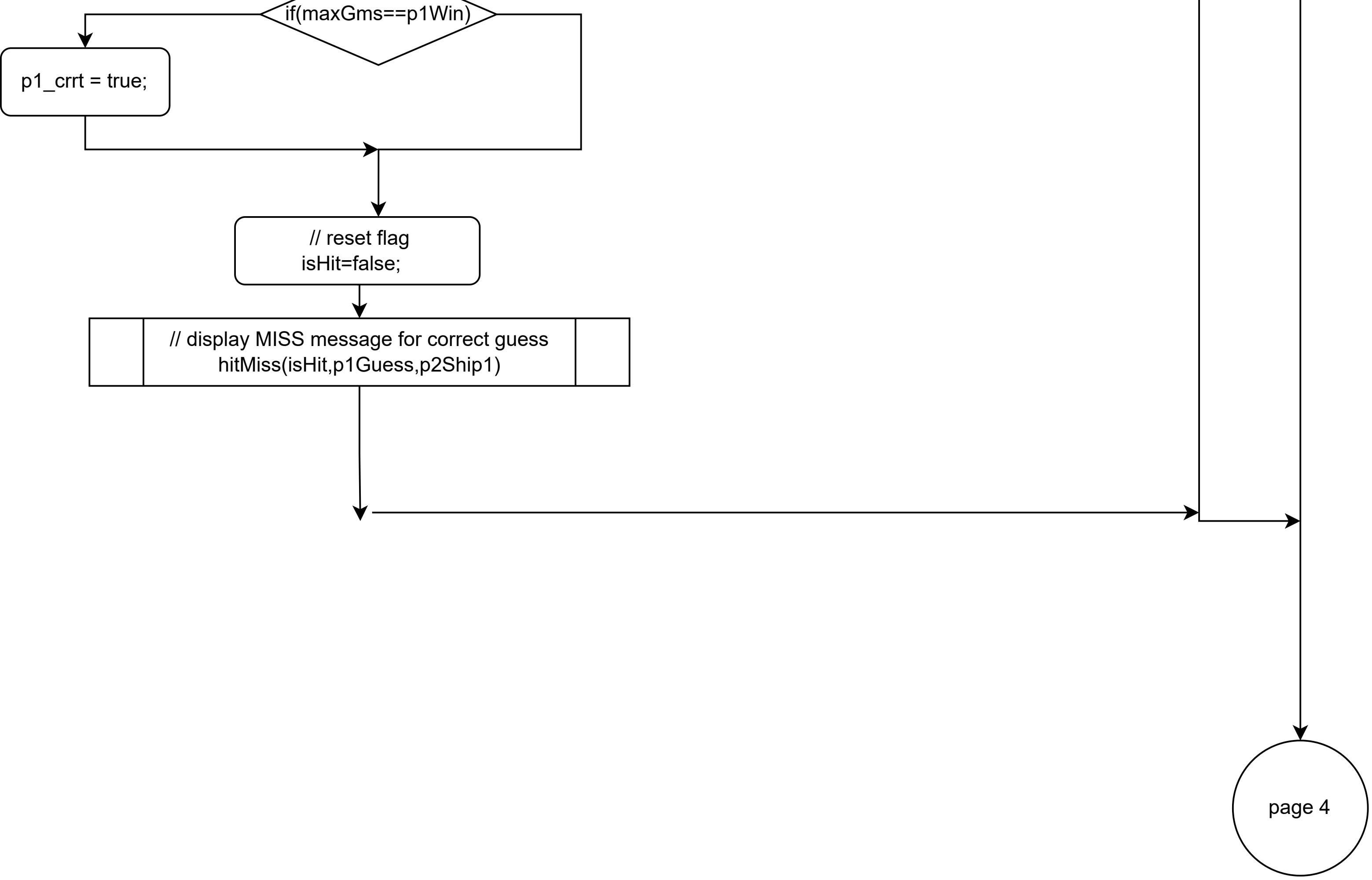
Battleship
FlowChart



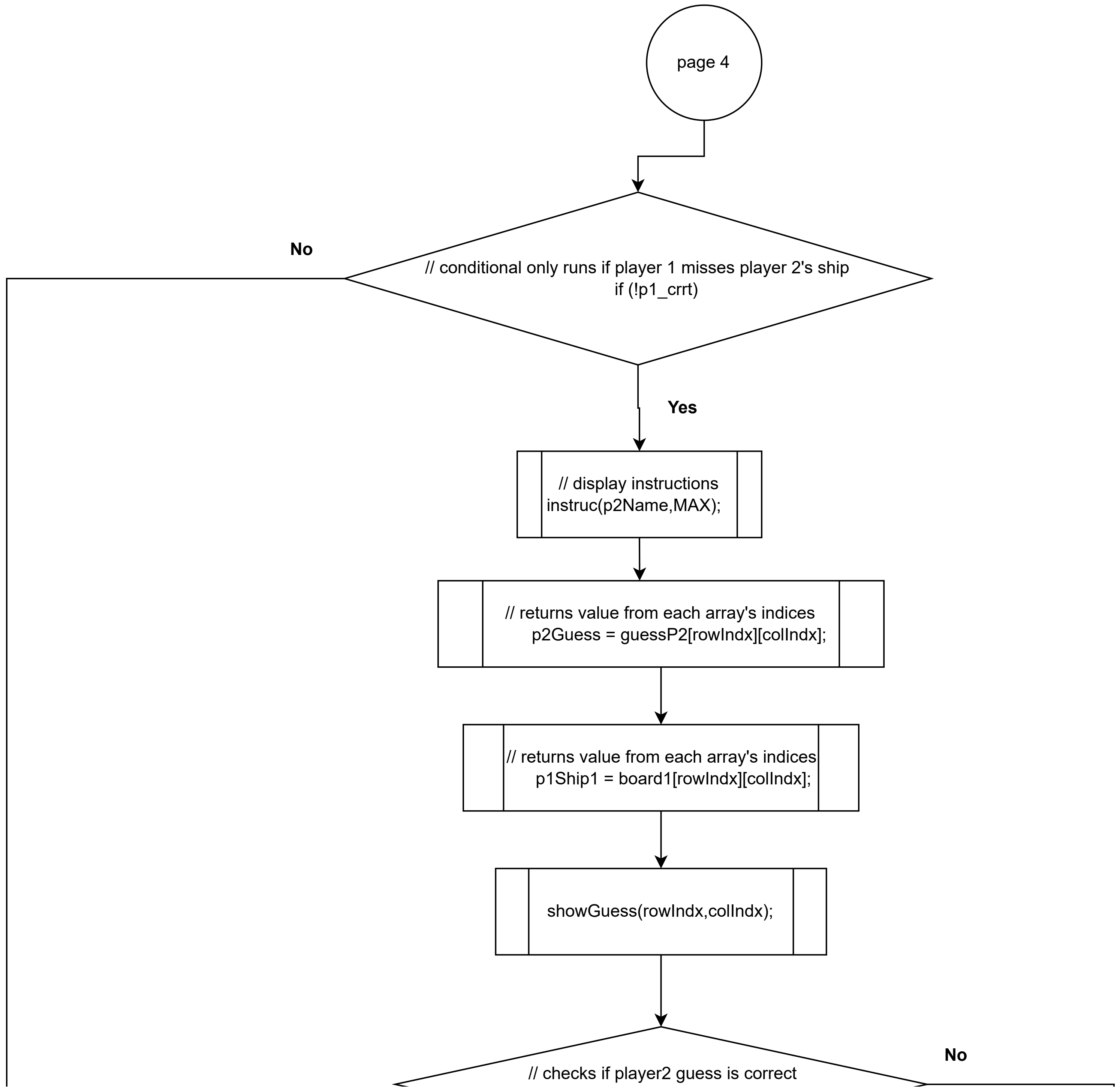
```
// ask for player 1's name
getName(p1Name)

// fill guess[] [] with choices[]
fllGArr(guessP1,guessP2,choices,ROWS,COLS,p1GShps,p2GShps);

runMenu(p1Name,ch,ans,count1,count2,numShp1,numShp2,guessP1,guessP2,
board1,board2, p1GShps,p2GShps,ROWS,COLS,maxGms,winner,p2Names,
SIZE7,names,SIZE8,vNames)

//pass by value
string gameNme = "BATTLE";

banner(gameNme);

// get player 2's name from an array of names
string p2Name=pickP2(p2Names,SIZE7);

cout << setw(12) << p1Name << " vs " <<
p2Name << "!" << endl;

pause();

// sets games flag & counting variables to their default values
p1_crrt = p2_crrt = false;
rowIndx=colIndx=1;
```

Battleship FlowChart

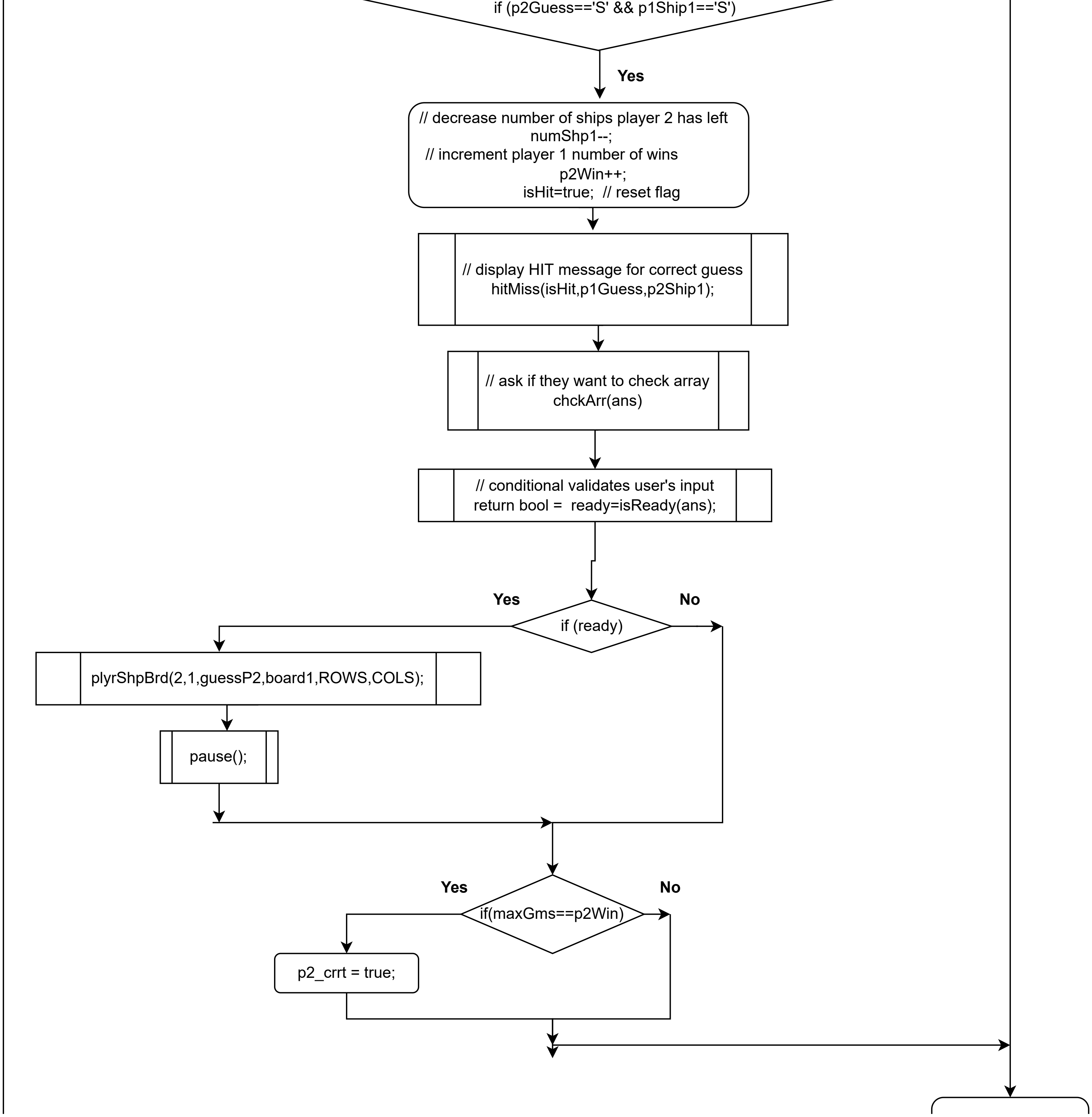

page 3
while(!p1_crrt) && (!p2_crrt))
rBanner(round);
instruc(p1Name,MAX);
// returns value from each array's indices
p1Guess = guessP1[rowIndx][colIndx];
// returns value from player's board
char p2Ship1 = board2[rowIndx][colIndx];
//show players guess
showGuess(rowIndx,colIndx);
// checks if player1 guess is correct
if (p1Guess=='S' && p2Ship1=='S')
numShp2-- // decrease number of ships player 2 has left
p1Win++ // increment player 1 number of wins
isHit=true; // reset flag
// display HIT message for correct guess
hitMiss(isHit,p1Guess,p2Ship1);
// ask if they want to check array
chckArr(ans)
// conditional validates user's input
return bool = ready=isReady(ans);
if(ready)
plyrShpBrd(1,2,guessP1,board2,ROWS,COLS);
pause();

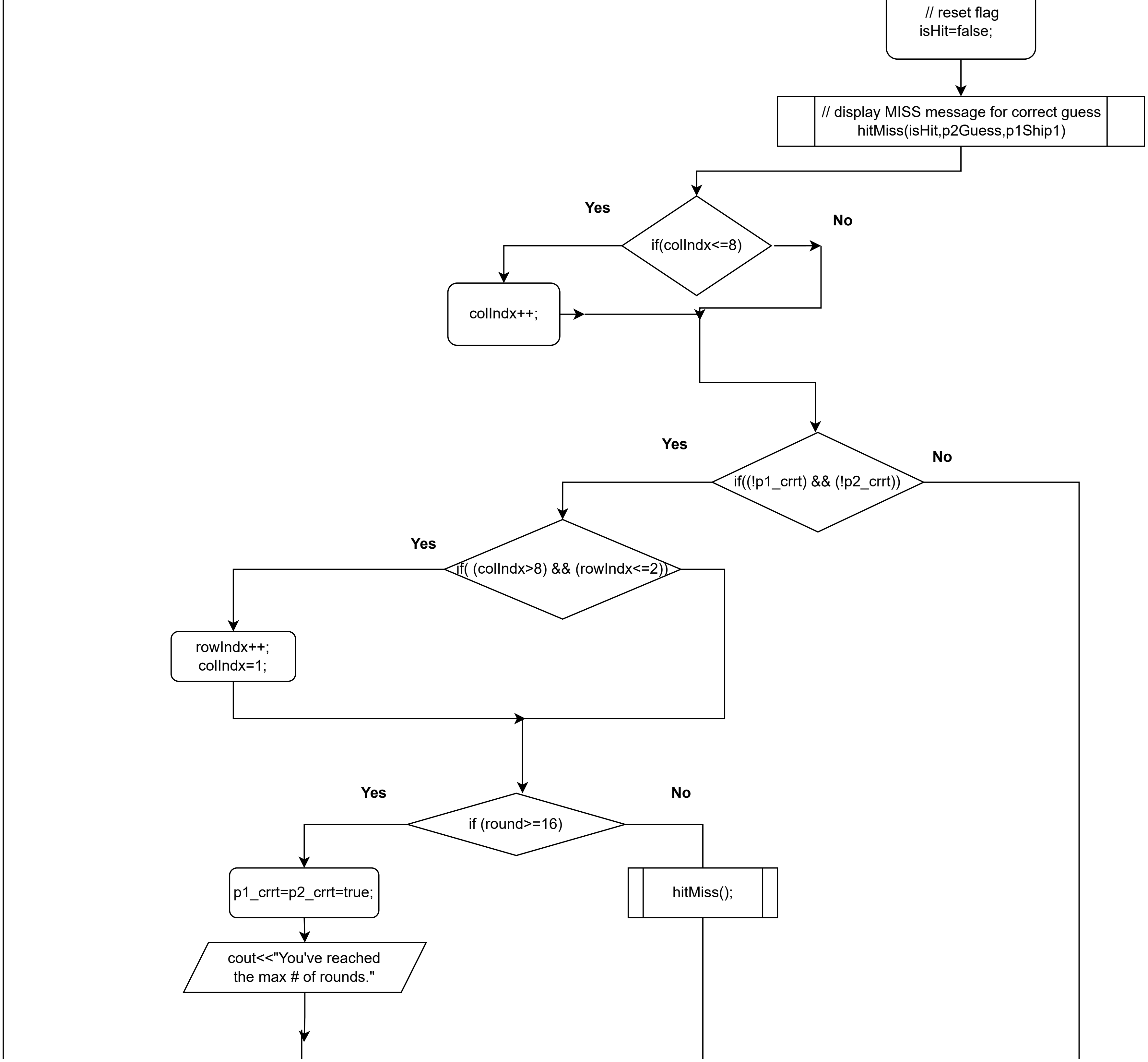

if(maxGms==p1Win)
p1_crrt = true;
// reset flag
isHit=false;
// display MISS message for correct guess
hitMiss(isHit,p1Guess,p2Ship1)
page 4

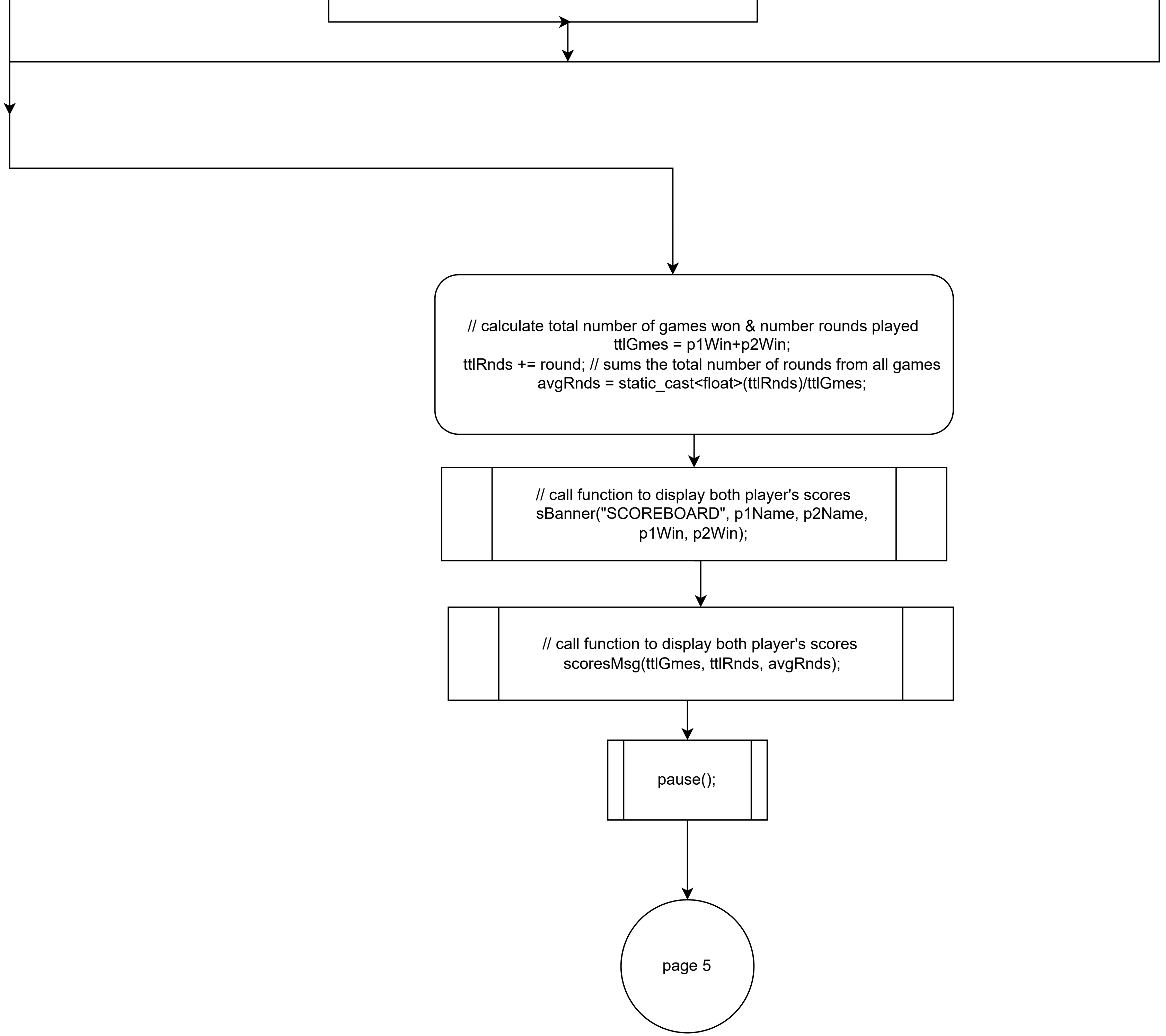
Battleship
FlowChart



```
// conditional only runs if player 1 misses player 2's ship
if (!p1_crrt)

// display instructions
instruc(p2Name,MAX);

// returns value from each array's indices
p2Guess = guessP2[rowIndx][colIndx];

// returns value from each array's indices
p1Ship1 = board1[rowIndx][colIndx];

showGuess(rowIndx,colIndx);

// checks if player2 guess is correct
```


if (p2Guess=='S' && p1Ship1=='S')
Yes
// decrease number of ships player 2 has left
numShp1--;
// increment player 1 number of wins
p2Win++;
isHit=true; // reset flag
// display HIT message for correct guess
hitMiss(isHit,p1Guess,p2Ship1);
// ask if they want to check array
chckArr(ans)
// conditional validates user's input
return bool = ready=isReady(ans);
Yes
No
if (ready)
plyrShpBrd(2,1,guessP2,board1,ROWS,COLS);
pause();
Yes
No
if(maxGms==p2Win)
p2_crrt = true;


// reset flag
isHit=false;
// display MISS message for correct guess
hitMiss(isHit,p2Guess,p1Ship1)
Yes
No
if(colIndx<=8)
colIndx++;
Yes
No
if((!p1_crrt) && (!p2_crrt))
Yes
if((colIndx>8) && (rowIndx<=2))
rowIndx++;
colIndx=1;
Yes
No
if (round>=16)
p1_crrt=p2_crrt=true;
hitMiss();
cout<<"You've reached
the max # of rounds."



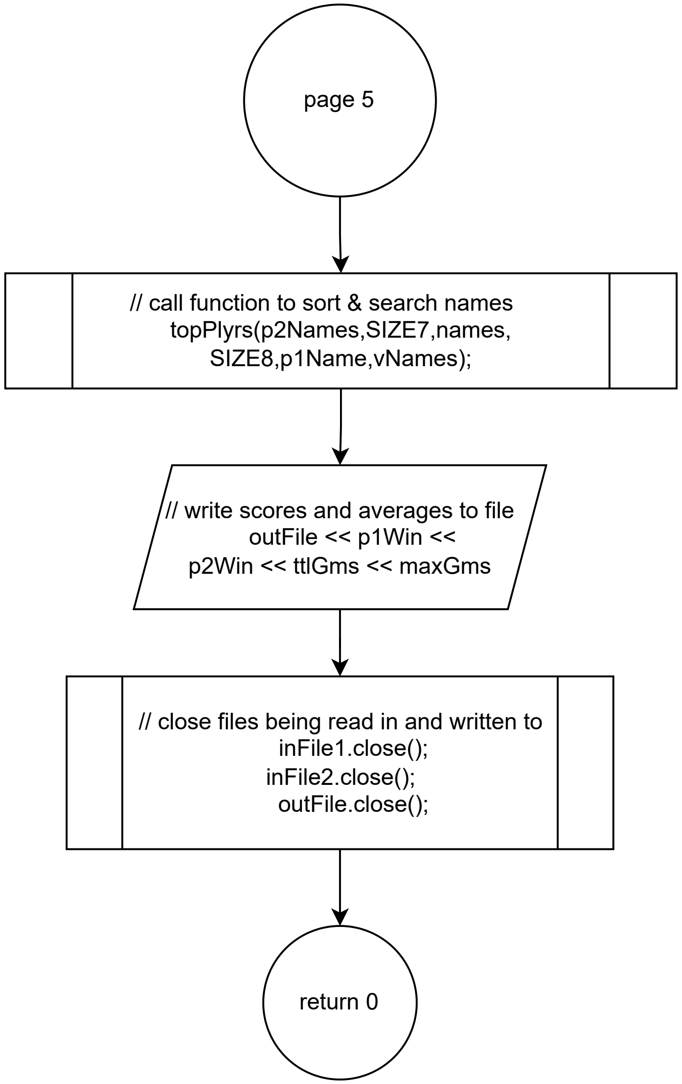
```
// calculate total number of games won & number rounds played
ttlGmes = p1Win+p2Win;
ttlRnds += round; // sums the total number of rounds from all games
avgRnds = static_cast<float>(ttlRnds)/ttlGmes;
```

```
// call function to display both player's scores
sBanner("SCOREBOARD", p1Name, p2Name,
p1Win, p2Win);
```

```
// call function to display both player's scores
scoresMsg(ttlGmes, ttlRnds, avgRnds);
```

```
pause();
```

Battleship FlowChart



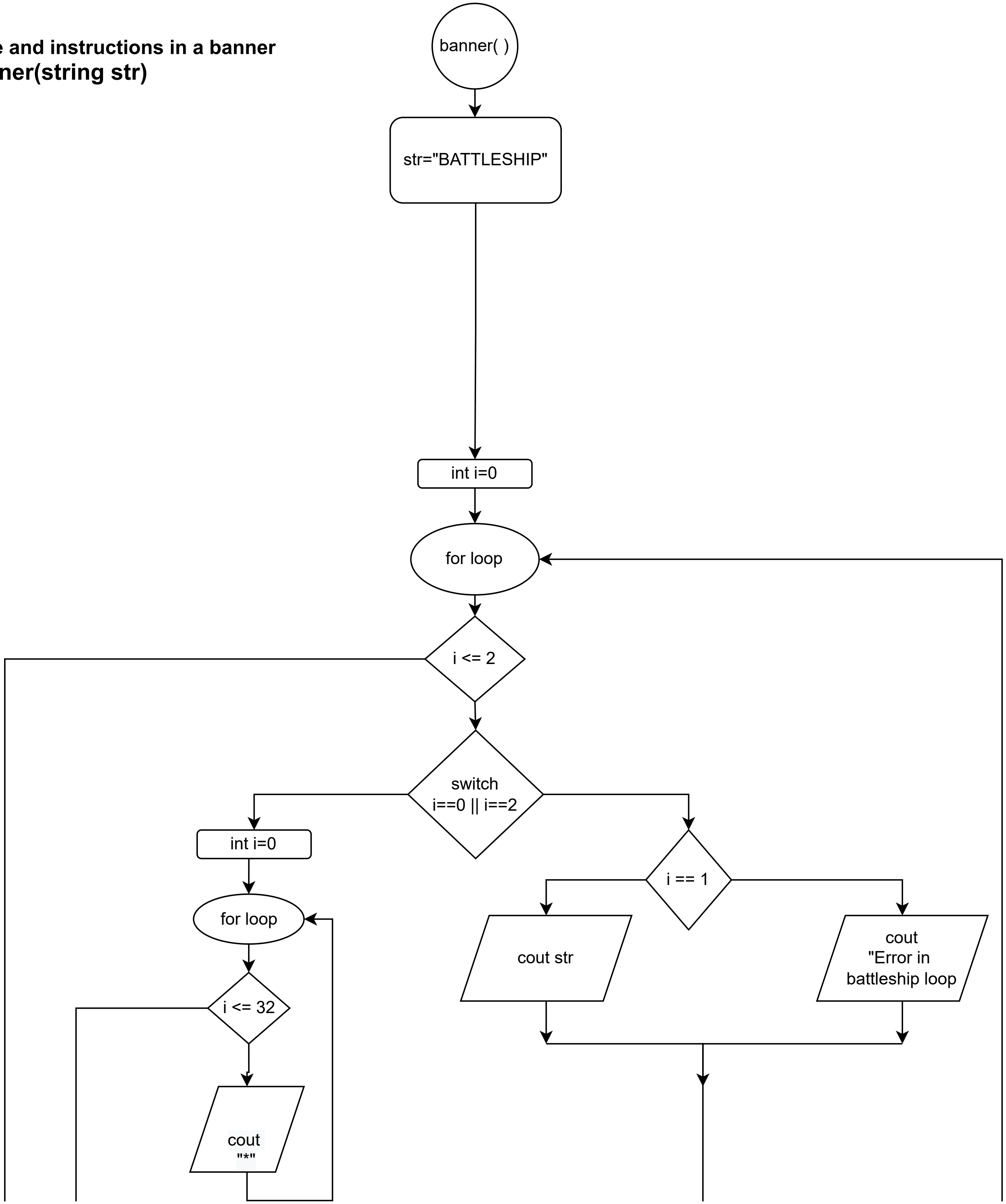
```
// call function to sort & search names
topPlyrs(p2Names,SIZE7,names,
SIZE8,p1Name,vNames);
```

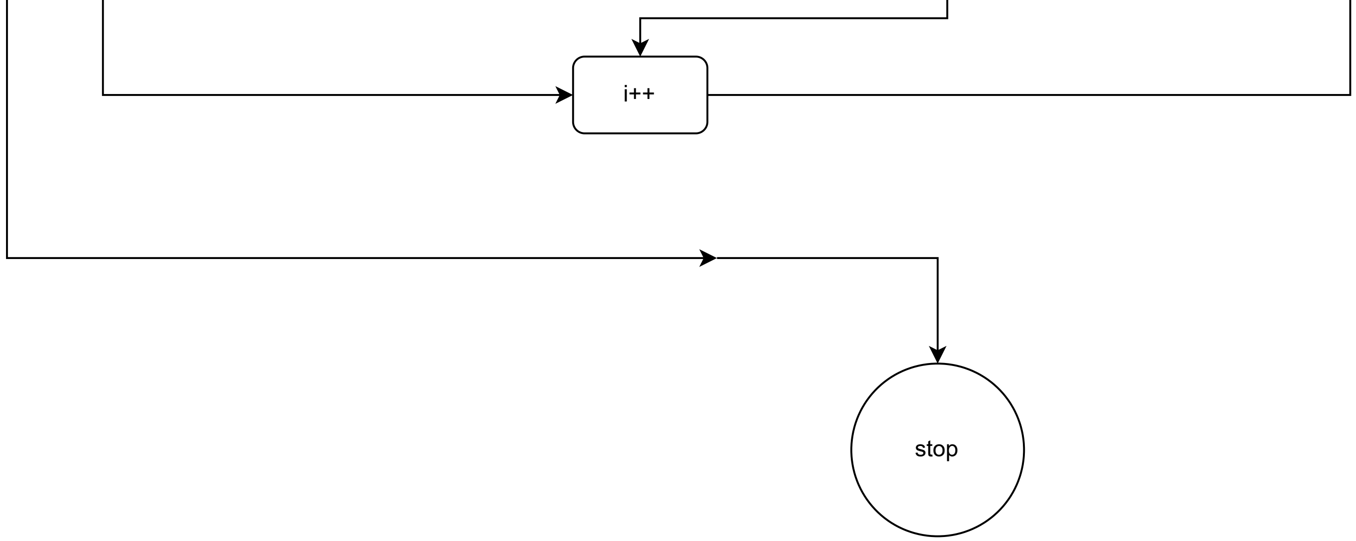
```
// write scores and averages to file
outFile << p1Win <<
p2Win << ttlGms << maxGms
```

```
// close files being read in and written to
inFile1.close();
inFile2.close();
outFile.close();
```

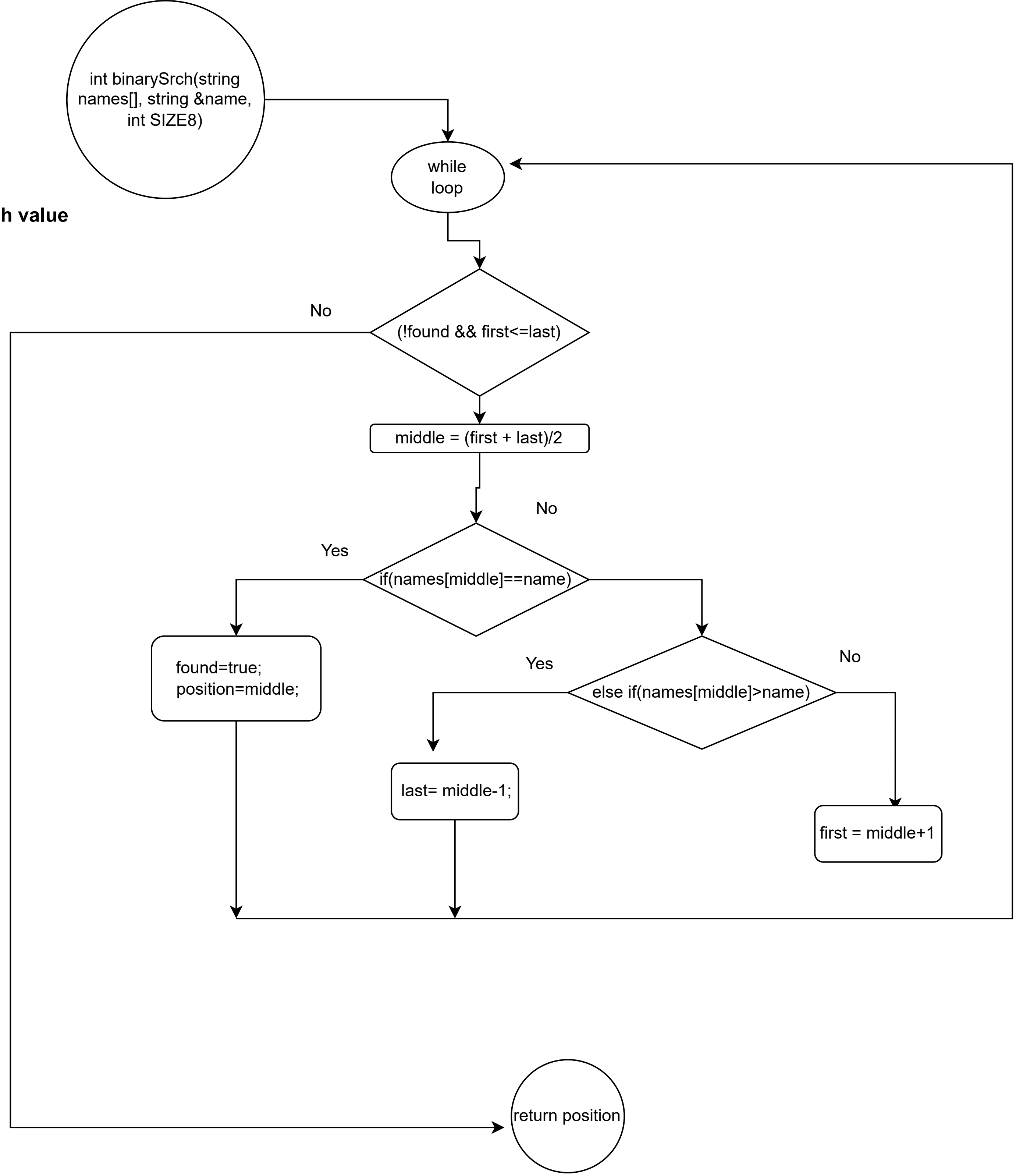
// displays game's name and instructions in a banner

void banner(string str)

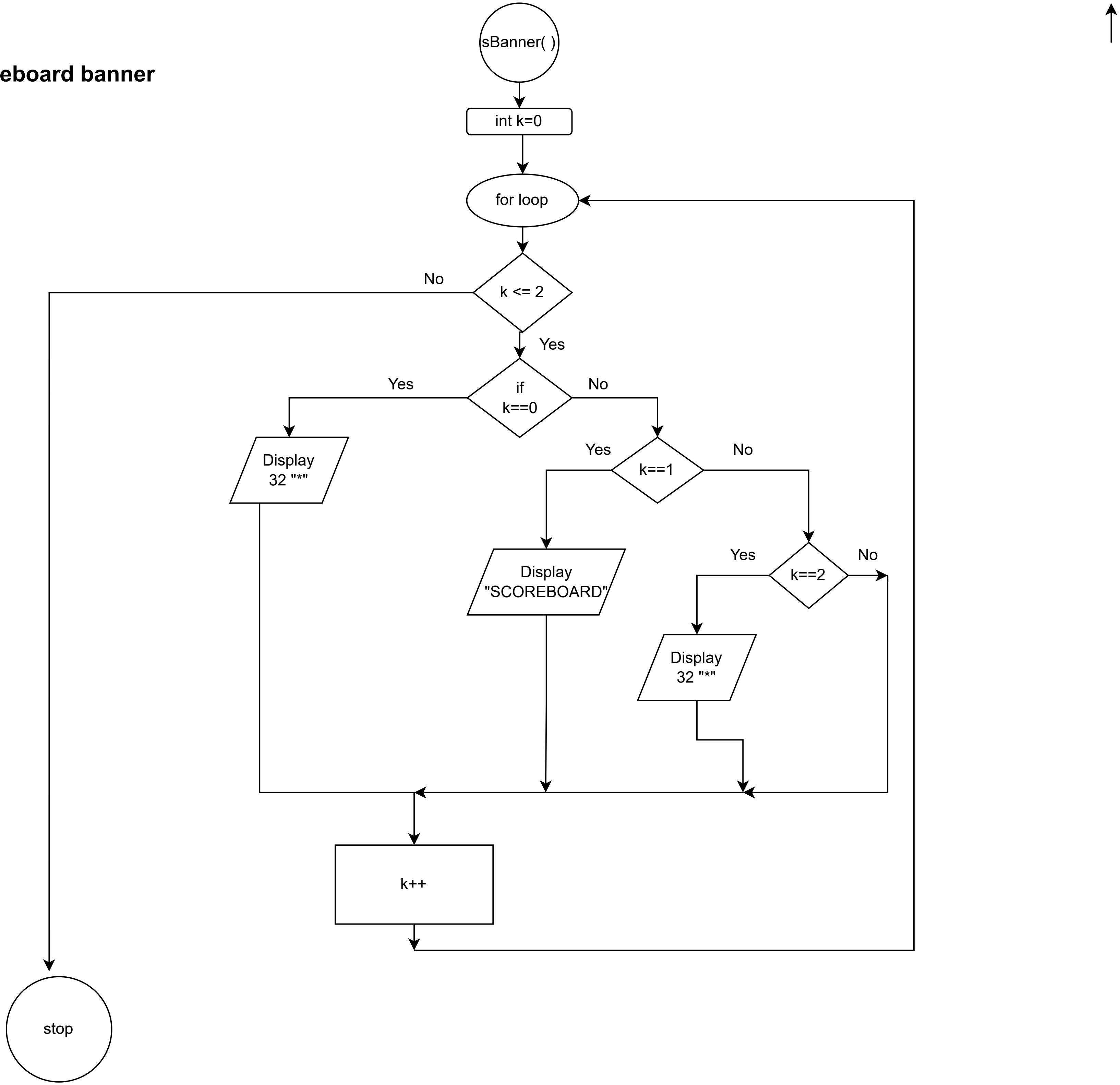



i++
stop

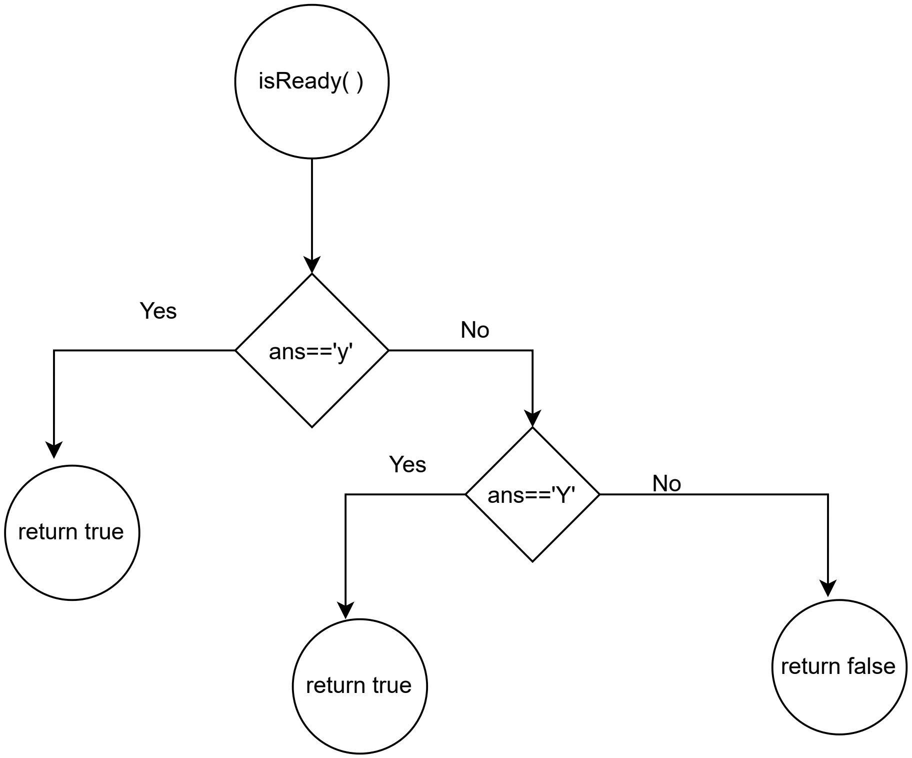
returns index position of search value

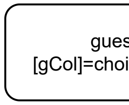


Scoreboard banner

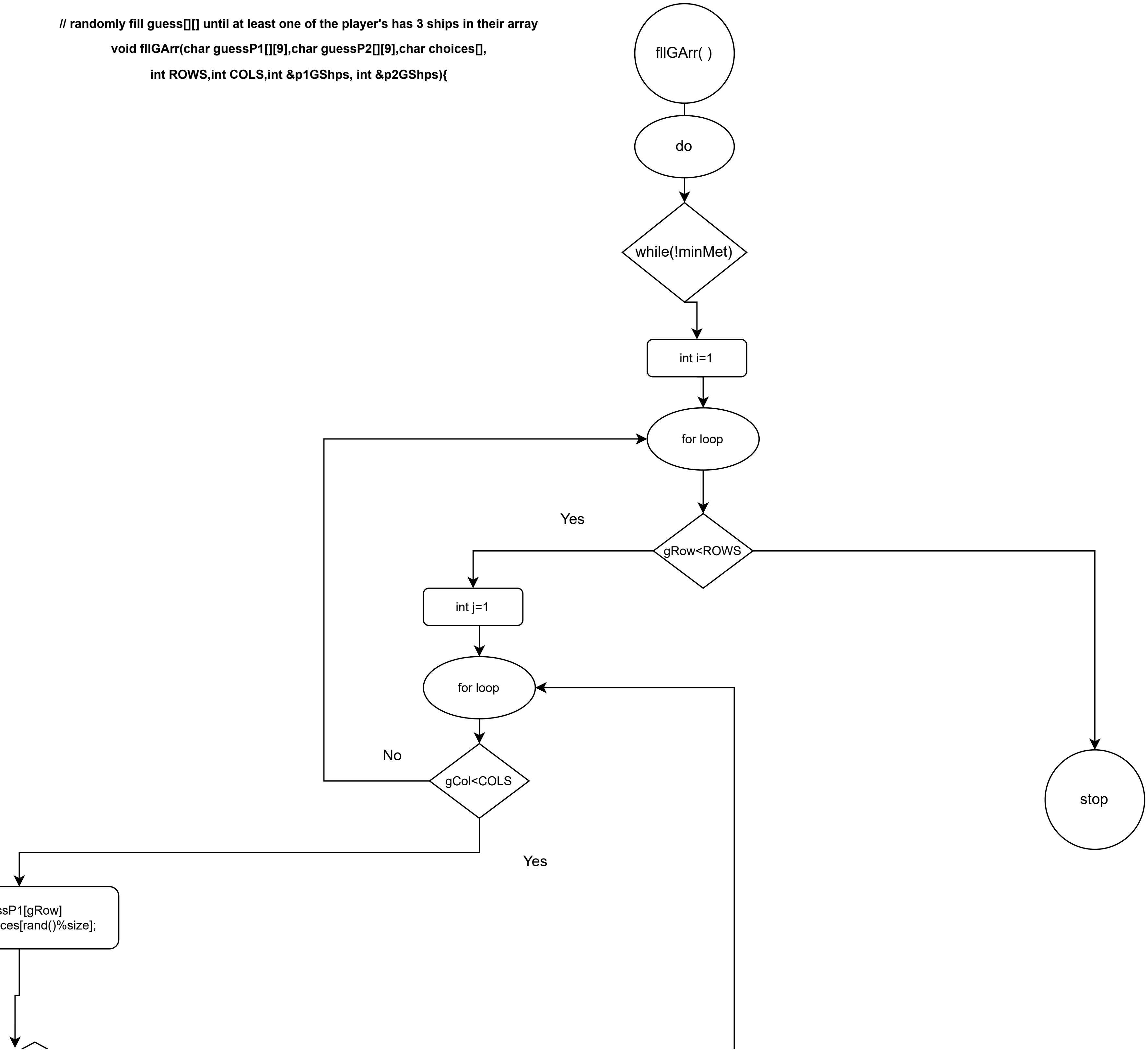


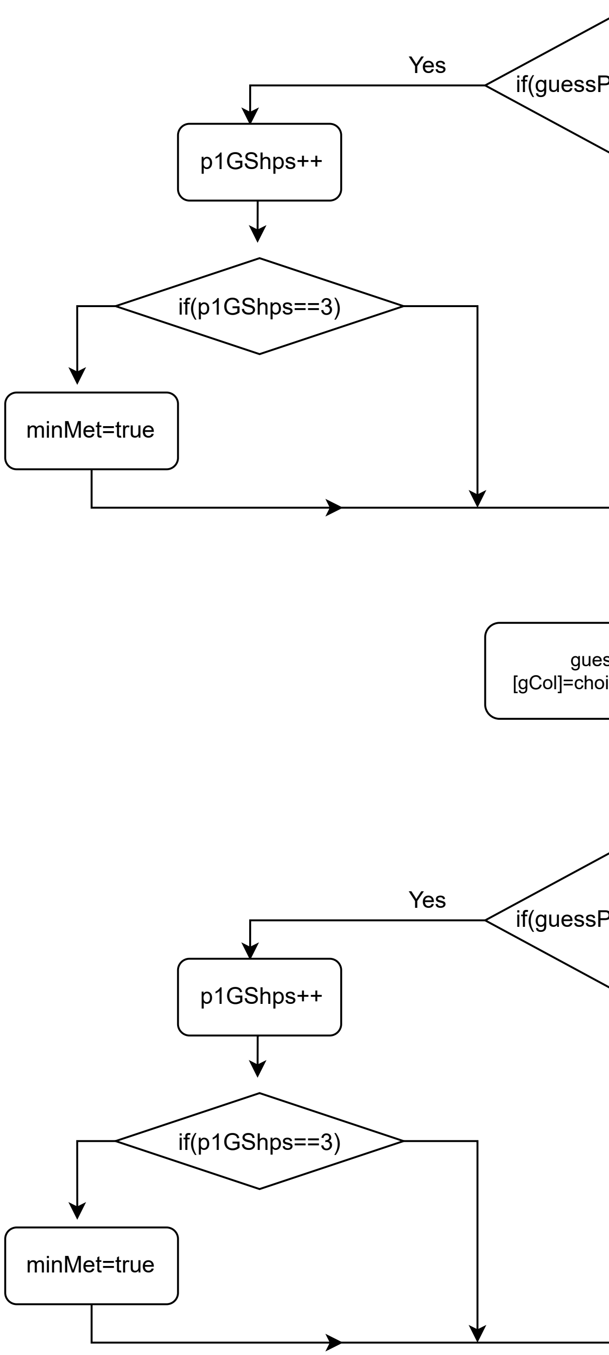
bool isReady(char ans)

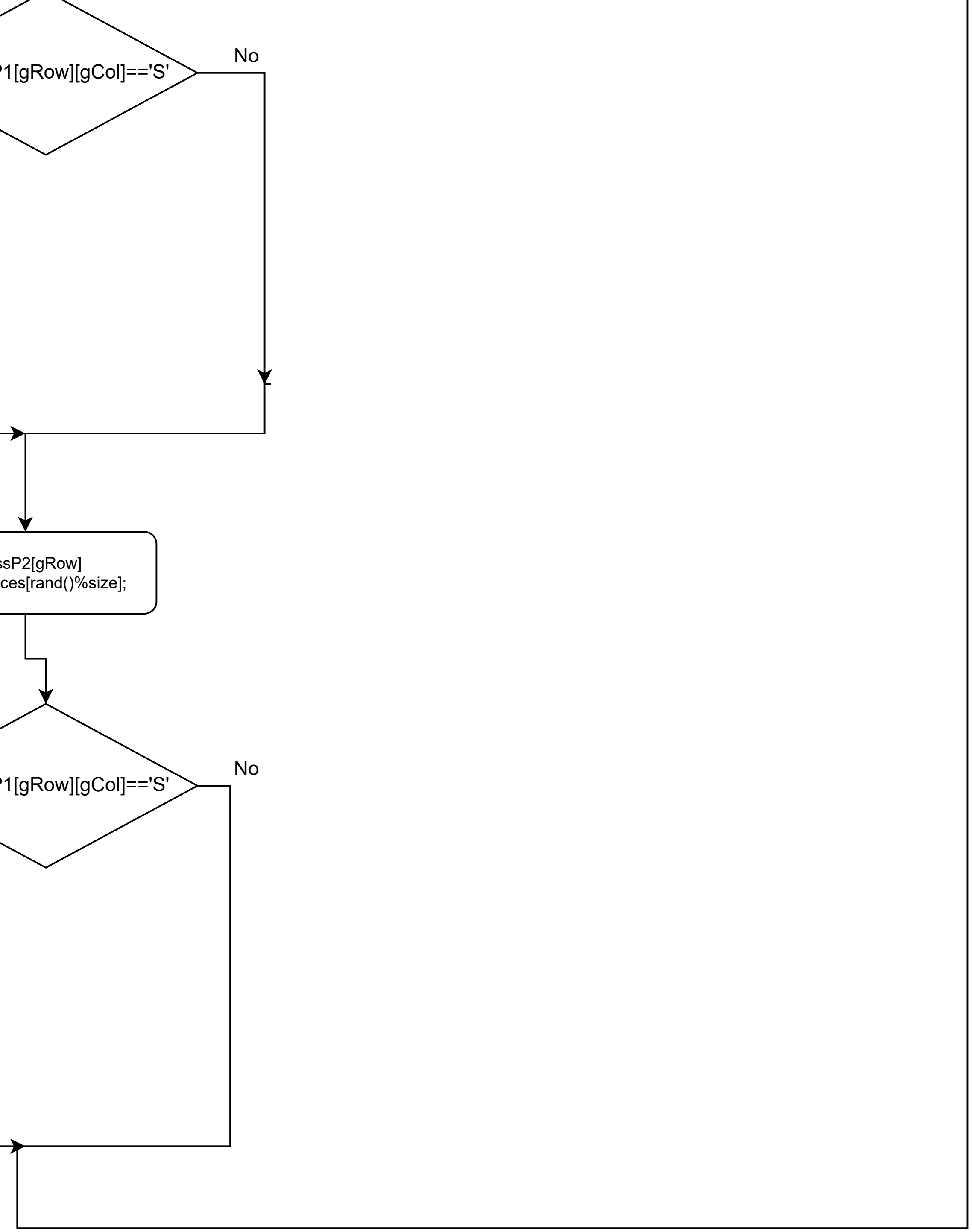



gues
[gCol]=choi

```
// randomly fill guess[][] until at least one of the player's has 3 ships in their array
void fllGArr(char guessP1[][9],char guessP2[][9],char choices[],
int ROWS,int COLS,int &p1GShps, int &p2GShps){
```




Yes
if(guessP
p1GShps++
if(p1GShps==3)
minMet=true
gues
[gCol]=choi
Yes
if(guessP
p1GShps++
if(p1GShps==3)
minMet=true



?1[gRow][gCol]=='S'
No
ssP2[gRow]
ces[rand()%size];
?1[gRow][gCol]=='S'
No