Step 5:

//Function for top bins

void showTopBins() {

cout << "\* ";//first bin

for (int t = 0; t < 6; t++) {

cout << setw(3); // spacing for numbers

cout << " \*"; // star between numbers

cout << setw(4); // spacing for numbers

//All bins start with 4

cout << 4 ; // all bins are 4

}

cout<< " \* \*";//last bin

}

//Function for bottom bins

void showBottomBins() {

cout << "\* 0 ";//first bin

for (int t = 0; t < 6; t++) {

cout << setw(3); // spacing for numbers

cout << " \*"; // star between numbers

cout << setw(4); // spacing for numbers

//All bins start with 4

cout << 4 ; // all bins have 4

}

cout<< " \* 0 \*";//last bin

}

//new showBoard modifications for bin numbers

//Function to display board

void showBoard () {

makeSolidLine(57); //top line

cout << endl;

makeDottedLine(); //columns

cout<< endl;

makeDottedLine();//columns

cout << endl;

showTopRowNumbers(); // Number bin

cout << endl;

makeDottedLine(); //columns

cout<< endl;

showTopBins(); //number of beads on top

cout << endl;

makeDottedLine();//columns

cout << endl;

makeDottedLine(); //columns

cout << endl;

cout << "\*"; //middle line first point

cout << " 13 "; //spacing for larger bin

makeSolidLine(43); // middle line

cout << " 6 \*"; //spacing for larger bin

cout << endl;

makeDottedLine(); //columns

cout<< endl;

makeDottedLine();//columns

cout << endl;

makeDottedLine(); //columns

cout<< endl;

showBottomRowNumbers(); // show bottom numbers

cout << endl;

makeDottedLine();//columns

cout << endl;

showBottomBins(); // number of beads in bottom

cout << endl;

makeDottedLine(); //columns

cout << endl;

makeDottedLine(); //columns

cout << endl;

makeSolidLine(57); // bottom line

}

Part 6:

//Game over Function

void gameOverCheck(int beadArray[MAX]) {

//vairables

int winner = 0 ;// no winner yet

int gameStatus = -1; //game not over

int side1 = 0; //top side

int side2= 0; //botom side

//Loops to check that there are no beads in bins

//top bins

for (int i =0; i<6; i++) {

side1 += beadArray[i];

}

//bottom bins

for (int i =12; i >6; i--) {

side2 += beadArray[i];

}

//Both sides are empty

if (side1 == 0 || side2 == 0) {

//check what side has the most beads (left or right)

beadArray[6]=beadArray[6]+side1;

beadArray[13]=beadArray[13]+side2;

if(beadArray[6]>beadArray[13]) {

winner = 1; // player 1 wins (collecting on right side)

}

else {

winner = 2; // player 2 wins (collecting on the left side)

}

else {

winner = -1;// no winner yet

}

return winner;

}