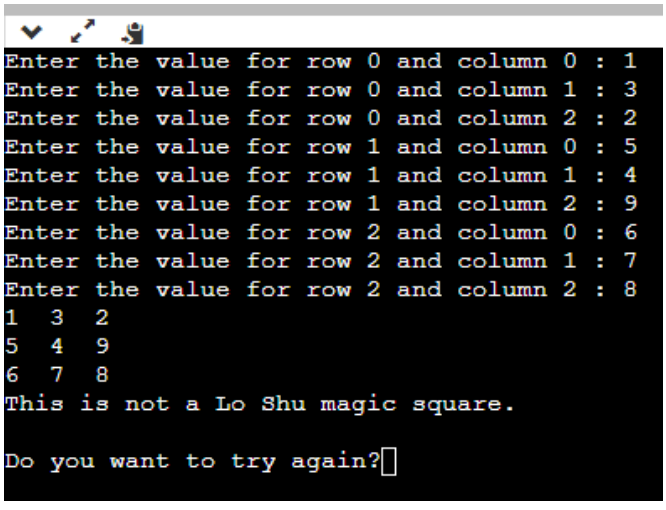
PROJECT 5

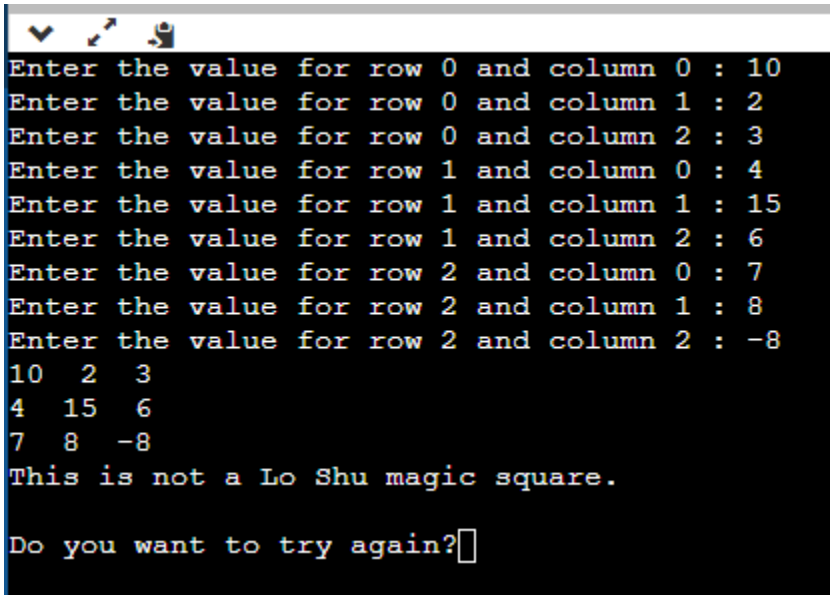
KIERAN YALLA

Test Case #	Input	Actual Input	Expected Output Is Magic?	Actual Output Is Magic?	Did the test pass?
1	1 3 2 5 4 9 6 7 8	1 3 2 5 4 9 6 7 8	No	No	Yes
2	10 2 3 4 15 6 7 8 -8	10 2 3 4 15 6 7 8 -8	No	No	Yes
3	4 9 2 3 5 7 8 1 6	4 9 2 3 5 7 8 1 6	Yes	Yes	Yes

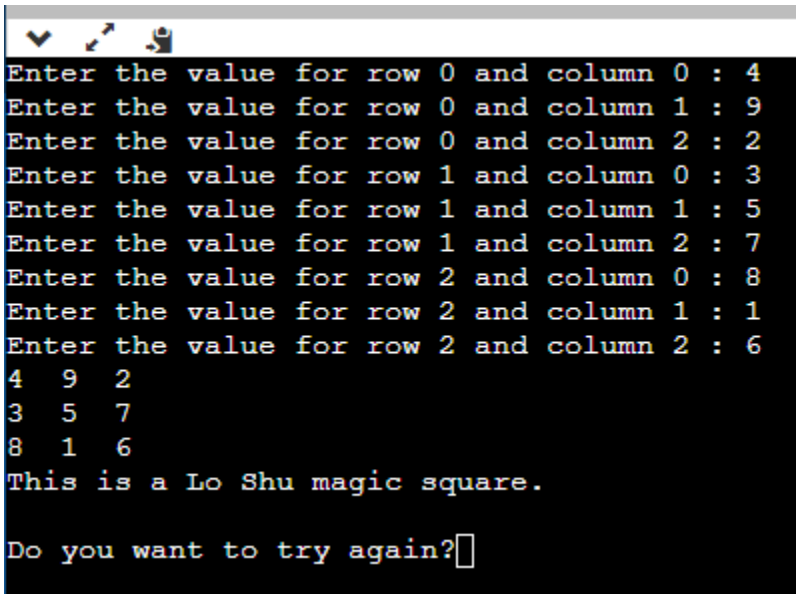
SAMPLE OUTPUT #1:



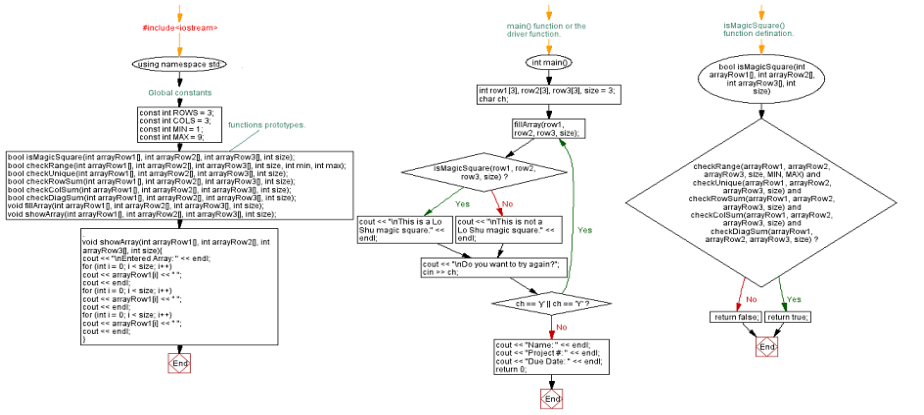
SAMPLE OUTPUT #2:

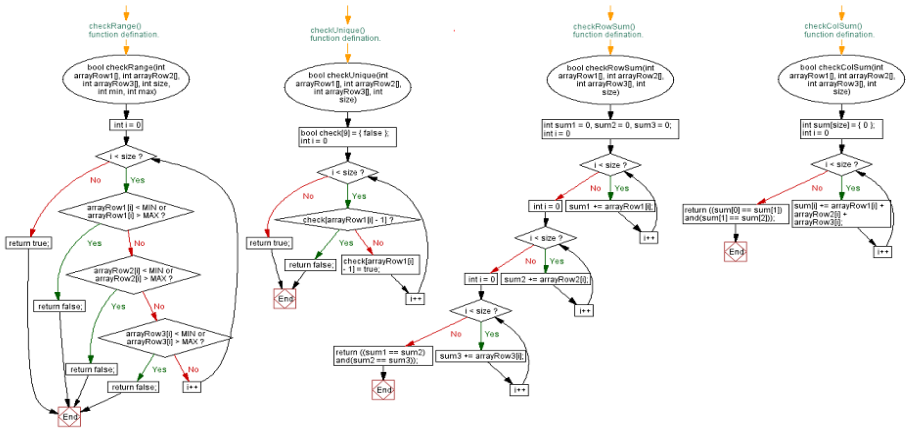


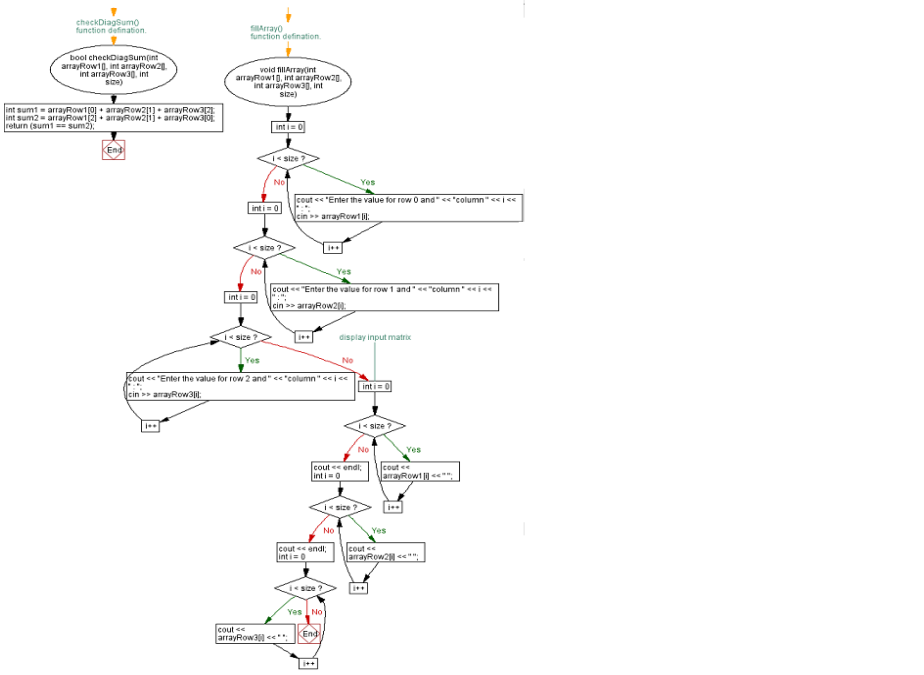
SAMPLE OUTPUT #3:



FLOWCHARTS:


#include<iostream>
using namespace std
Global constants
const int ROWS = 3;
const int COLS = 3;
const int MIN = 1;
const int MAX = 9;
functions prototypes.
bool isMagicSquare(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size);
bool checkRange(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size, int min, int max);
bool checkUnique(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size);
bool checkRowSum(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size);
bool checkColSum(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size);
bool checkDiagSum(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size);
void fillArray(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size);
void showArray(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size);
void showArray(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size){
cout << "\nEntered Array:" << endl;
for (int i = 0; i < size; i++)
cout << arrayRow1[i] << " ";
cout << endl;
for (int i = 0; i < size; i++)
cout << arrayRow1[i] << " ";
cout << endl;
for (int i = 0; i < size; i++)
cout << arrayRow1[i] << " ";
cout << endl;
}
End
main() function or the driver function.
int main()
int row1[3], row2[3], row3[3], size = 3;
char ch;
fillArray(row1, row2, row3, size);
isMagicSquare(row1, row2, row3, size) ?
Yes
cout << "\nThis is a Lo Shu magic square." << endl;
No
cout << "\nThis is not a Lo Shu magic square." << endl;
cout << "\nDo you want to try again?";
cin >> ch;
ch == 'y' || ch == 'Y' ?
Yes
No
cout << "Name:" << endl;
cout << "Project #:" << endl;
cout << "Due Date:" << endl;
return 0;
End
isMagicSquare()
function defination.
bool isMagicSquare(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size)
checkRange(arrayRow1, arrayRow2, arrayRow3, size, MIN, MAX) and checkUnique(arrayRow1, arrayRow2, arrayRow3, size) and checkRowSum(arrayRow1, arrayRow2, arrayRow3, size) and checkColSum(arrayRow1, arrayRow2, arrayRow3, size) and checkDiagSum(arrayRow1, arrayRow2, arrayRow3, size) ?
No
Yes
return false;
return true;
End


checkRange()
function defination.
bool checkRange(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size, int min, int max)
int i = 0
i < size ?
No
Yes
arrayRow1[i] < MIN or arrayRow1[i] > MAX ?
return true;
Yes
No
arrayRow2[i] < MIN or arrayRow2[i] > MAX ?
return false;
Yes
No
arrayRow3[i] < MIN or arrayRow3[i] > MAX ?
return false;
Yes
No
return false;
i++
End
checkUnique()
function defination.
bool checkUnique(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size)
bool check[9] = { false };
int i = 0
i < size ?
No
Yes
check[arrayRow1[i] - 1] ?
return true;
Yes
No
return false;
check[arrayRow1[i] - 1] = true;
End
i++
checkRowSum()
function defination.
bool checkRowSum(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size)
int sum1 = 0, sum2 = 0, sum3 = 0;
int i = 0
i < size ?
No
Yes
int i = 0
sum1 += arrayRow1[i];
i++
i < size ?
No
Yes
int i = 0
sum2 += arrayRow2[i];
i++
i < size ?
No
Yes
return ((sum1 == sum2) and(sum2 == sum3));
sum3 += arrayRow3[i];
i++
End
checkColSum()
function defination.
bool checkColSum(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size)
int sum[size] = { 0 };
int i = 0
i < size ?
No
Yes
return ((sum[0] == sum[1]) and(sum[1] == sum[2]));
sum[i] += arrayRow1[i] + arrayRow2[i] + arrayRow3[i];
i++
End


checkDiagSum()
function defination.
bool checkDiagSum(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size)
int sum1 = arrayRow1[0] + arrayRow2[1] + arrayRow3[2];
int sum2 = arrayRow1[2] + arrayRow2[1] + arrayRow3[0];
return (sum1 == sum2);
End
fillArray()
function defination.
void fillArray(int arrayRow1[], int arrayRow2[], int arrayRow3[], int size)
int i = 0
i < size ?
No
Yes
int i = 0
cout << "Enter the value for row 0 and " << "column " << i << " : ";
cin >> arrayRow1[i];
i++
i < size ?
No
Yes
int i = 0
cout << "Enter the value for row 1 and " << "column " << i << " : ";
cin >> arrayRow2[i];
i++
i < size ?
No
Yes
int i = 0
cout << "Enter the value for row 2 and " << "column " << i << " : ";
cin >> arrayRow3[i];
i++
display input matrix
i < size ?
No
Yes
cout << endl;
int i = 0
cout << arrayRow1[i] << " ";
i++
i < size ?
No
Yes
cout << endl;
int i = 0
cout << arrayRow2[i] << " ";
i++
i < size ?
Yes
No
cout << arrayRow3[i] << " ";
End
i++

PSEUDO CODE:

MagicSqaure()

arrayRow1 = input

arrayRow2 = input

arrayRow3 = input

ans= isMagicSquare(arrayRow1, arrayRow2,arrayRow3)

if checkRange() //checking whether all numbers are in the range or not

and checkUnique()// checking whether all numbers are unique in the square

and checkRowSum()// checking whether all rows have the same sum

and checkColumnSum()// checking whether all columns have the same sum

and checkDiagSum() // checking whether all the diagonals have the same sum

return true

return false

if ans is true : print("It is a magic square")

else : print("It is not a magic square")

END

Lessons Learned:

I learned a better flowchart maker than the one I had. It turns my code into a flowchart for me.