

# Laboratory 07A

CS-102

# Lo Shu Magic Square

- The Lo Shu Magic Square is a grid with 3 rows and 3 columns shown below. The Lo Shu Magic Square has the following properties:
  - The grid contains the numbers 1 through 9 exactly.
  - The sum of each row, each column, and each diagonal all add up to the same number (see below).

4	9	2
3	5	7
8	1	6

4	9	2	↗ 15
3	5	7	→ 15
8	1	6	→ 15
↓ 15	↓ 15	↓ 15	↘ 15

# Laboratory 07A

- In a program, you can simulate a magic square using a two-dimensional array.
- Write a function that accepts a two-dimensional array as an argument, and determines whether the array is a Lo Shu Magic Square or not.
- Test the function in a program.
- Hint: Before you begin, it would be helpful for you to answer the following questions.
  - After you've answered these questions, call the instructor over and ask if you've got answers to these questions that will be helpful to you in programming this problem.
  - Once the instructor has signed off on your answers, then you can proceed with your coding.

# Laboratory 07A – Part 1

- Question 1: What is the constant unchanging number of rows, and number of columns that the array that you will use, will always have?  
\_\_\_\_\_
- Question 2: What is the sum that the 3 elements in a row, a column, or a diagonal will always add up to? \_\_\_\_\_
- Question 3: How many different tests (i.e. if statements) will you have to do to determine if the square is a Lo Shu square?  
\_\_\_\_\_
- If you are taking this Lab synchronously, show the instructor your answers to the above questions.
- If you are taking this Lab asynchronously, answer the above questions in a text file or word file called: *YourName-Lab07A-1.txt* or *YourName-Lab07A-1.docx*, and submit this file to Canvas.

# Laboratory 07A – Part 2

```
// This is your Lo Shu Magic Square Detector Template
// You only need to write the function at the bottom (see next page)
#include <iostream>
using namespace std;

const int CORRECTSUM = 15;
const int SIZE = 3;
int loShu[SIZE][SIZE] = {{0, 0, 0},
                          {0, 0, 0},
                          {0, 0, 0}};

bool isLoShu();      // Prototype
```

# Laboratory 07A – Part 2

```
int main()
{
    for (int i = 0; i < SIZE; i++)
        for (int j = 0; j < SIZE; j++)
        {
            cout << "Enter element for row " << i+1 << ", column " << j+1 << ": ";
            cin >> loShu[i][j];
        }
    cout << (isLoShu() ? "It's a Lo Shu!" : "It's not a Lo Shu!") << endl;
    return 0;
}

bool isLoShu()
{
    // Your job is to write what goes into this function right here.
}
```

# Laboratory 07A – Part 2

- Call your program *YourName-Lab07A-2.cpp*.
- If you are taking this Lab synchronously, demonstrate your Lo Shu Magic Square Detector program to the instructor.
- If you are taking this Lab asynchronously, submit your *YourName-Lab07A-2.cpp* to Canvas.

# Laboratory 07A – Part 3

- Shown at the beginning of the Lab are the values of a true Lo Shu Magic Square. Can you think of a different arrangement of numbers that will also result in a Lo Shu Magic Square?
- Test out your new Lo Shu Magic Square Candidate with your program, from part 2, to make sure that it works.
- If you are taking this Lab synchronously, show your different Lo Shu Magic Square to the Lab instructor.
- If you are taking this Lab asynchronously, submit your different Lo Shu Magic Square in a text document or Word document to Canvas. Call your document: *YourName-Lab07A-3.txt* or *YourName-Lab07A-3.docx*.



# Laboratory 07A – Part 4

- How many different arrangements of a Lo Shu Magic Square are there?
- List as many 3 by 3 Lo Shu Squares as you are able.
- If you are doing this Lab synchronously, show your Lo Shu Magic Squares to the Instructor.
- If you are doing this Lab asynchronously, submit a text file or Word document containing all the Lo Shu arrangements you were able to come up with. Call your document: *YourName-Lab07A-4.txt* or *YourName-Lab07A-4.docx*.

# Laboratory 07A – Part 5

## Extra Credit

- Write a program that generates all of the Lo Shu Magic Squares.
  - hint: there aren't that many!
- Call your program *YourName*-Lab07A-5.cpp.
- If you are doing this Lab synchronously, demonstrate your program to the Instructor.
- If you are doing this Lab asynchronously, submit your program to canvas.