CS1 Spring 2017

Assignment 2: Magic Squares

A Magic Square of order N consists of N2 squares, each of which contains a number from 1 through N2 such that the rows, columns, and diagonals sum to the same thing. For example, the following is a magic square of order 3:

|  |  |  |
| --- | --- | --- |
| 6 | 1 | 8 |
| 7 | 5 | 3 |
| 2 | 9 | 4 |

Your program should ask the user for an **odd** number greater than or equal to 3 and print the magic square for that order. It should reject even numbers with a message that the user must enter an odd number or zero to exit. That is, your program should request a number, print the magic square, then request another number, and print that magic square. The program ends when the user enters zero for the order. You don’t need to print the boxes around the numbers, but arrange the numbers in columns so the square is readable, using print formatting. After printing each square, print the magic sum. For the square above, it is 15. That is, the rows, columns, and diagonals sum to 15.

Obviously, to solve this problem you will have to create a two-dimensional array. You should allocate this array dynamically, not have an array declared in your program. This makes the program more flexible.

**Hand in your .CPP file through eLearning.**

|  |  |
| --- | --- |
| **Grading criteria:** | |
| Program runs correctly for good data | 45 |
| Program is structured well | 40 |
| Program uses dynamic arrays for the squares | 5 |
| Program comments | 10 |

Request input and validating

Create a dynamic array to hold the square