

C Language Programming: Homework #3
Assigned on 10/22/2013, Due on 10/29/2013

Write a program that prints an $n \times n$ magic square (a square arrangement of the numbers 1, 2, ..., n^2 in which the sums of the rows, columns, and diagonals are all the same). For example, the user run the program by specifying the value of n :

This program creates a magic square of a specified size.

The size must be an *odd* number between 1 and 99.

Enter size of magic square: 5

17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 25 2 9

The diagram shows a 5x5 magic square with numbers 1 through 25. Arrows indicate the path of numbers starting from 1 in the center (row 2, column 2). The path moves up and right, wrapping from row 0 to row 4 and from column 4 to column 0. The path ends at 25 in the bottom row, column 2.

Store the magic square in a two-dimensional array. Start by placing the number 1 in the middle of row 0. Place each of the remaining numbers 2, 3, ..., n^2 by moving up one row and right one column. Any attempt to go outside the bounds of the array should “wrap around” to the opposite side of the array. For example, instead of storing the next number in row -1 , we would store it in row $n - 1$ (the last row). Instead of storing the next number in column n which does not exist, we would store it in column 0. If a particular array element is already occupied, put the number directly below the previously stored number. Try the following two options for array in two different files like hw3a.c and hw3b.c:

- (1) Use traditional static array like `magic[99][99]`.
- (2) Use variable-length arrays of n rows and n columns.

Also, you have to use the following functions that must have appropriate parameters to pass the 2D array in:

```
void create_magic_square (...);  
void print_magic_square (...);
```

Note: Although it is not required, you can output the results to a FILE and use command line to input n .