Magic Square

Writeup by **Vincent Chan**, id **815909699**.

Overview

Write a program that generates magic squares of an odd order. The program will take any odd number above 1 and generate a magic square using the following rules:

1. 1 will be places at the top row, in the middle.
2. Every number after 1 will be place one row up, one column right. The next number is k+1. In this case, 2.
3. If a number goes past the top of the rows and the *j*th column, place that number on the bottom row and the *j*th column.
4. if a number goest past the last column in the nth row, place that number on the leftest column and the nth row.
5. if a number goes past **both** the top row and the last column, place that number under the last number placed. If the space the number to be placed is occupied, place the number under the last number placed.

magic_square.cpp

```
#include <iostream>
#include <iomanip>

using namespace std;

main () {
  //This will ask for user input and
  //verify that it is a valid input.
  int order;
  cout << "Magic Square Generator\n" << "====================\n";
  cout << "Enter Order of square: ";
  cin >> order;
  if(order==1 | order%2==0) {
    cout << "Invalid number, please enter an odd number greater than 1.\n";
    return 0;
  }

  //This will initialize the square, zeroing out the elements.
  int square[order][order];
  for(int i=0; i<order; i++)
    for(int n=0; n<order; n++)
      square[i][n] = 0;
  square[0][order/2] = 1;
  int nextRow = -1;
  int nextColumn = (order/2)+1;

  //This will populate the array,
  //while also checking for special cases and adjusting.
  for(int nextInt=2; nextInt<=(order*order); nextInt++) {
    if(nextColumn>=order && nextRow<0) {
      nextColumn -= 1;
      nextRow += 2;
    }
    if(nextColumn>=order)
      nextColumn = 0;
    if(nextRow<0)
      nextRow = order-1;
    if(square[nextRow][nextColumn]!=0) {
      nextColumn -= 1;
      nextRow +=2;
    }
    square[nextRow][nextColumn] = nextInt;
    nextRow--;
    nextColumn++;
  }

  //This prints the array once the generation is complete.
  for(int i=0; i<order; i++) {
    for(int n=0; n<order; n++) {
      cout << setw(5);
      cout << square[i][n];
    }
    cout << endl;
  }
} //End main
```

Test Cases

This program was tested on 4 orders:

Order 5

```
Magic Square Generator
======================
Enter Order of 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
```

Order 7

```
Magic Square Generator
======================
Enter Order of square: 7
   30   39   48    1   10   19   28
   38   47    7    9   18   27   29
   46    6    8   17   26   35   37
    5   14   16   25   34   36   45
   13   15   24   33   42   44    4
   21   23   32   41   43    3   12
   22   31   40   49    2   11   20
```

Order 9

```
Magic Square Generator
======================
Enter Order of square: 9
   47   58   69   80    1   12   23   34   45
   57   68   79    9   11   22   33   44   46
   67   78    8   10   21   32   43   54   56
   77    7   18   20   31   42   53   55   66
    6   17   19   30   41   52   63   65   76
   16   27   29   40   51   62   64   75    5
   26   28   39   50   61   72   74    4   15
   36   38   49   60   71   73    3   14   25
   37   48   59   70   81    2   13   24   35
```

Order 11

```
Magic Square Generator
======================
Enter Order of square: 11
   68   81   94  107  120    1   14   27   40   53   66
   80   93  106  119   11   13   26   39   52   65   67
   92  105  118   10   12   25   38   51   64   77   79
  104  117    9   22   24   37   50   63   76   78   91
  116    8   21   23   36   49   62   75   88   90  103
```

```
   7   20   33   35   48   61   74   87   89  102  115
  19   32   34   47   60   73   86   99  101  114    6
  31   44   46   59   72   85   98  100  113    5   18
  43   45   58   71   84   97  110  112    4   17   30
  55   57   70   83   96  109  111    3   16   29   42
  56   69   82   95  108  121    2   15   28   41   54
```

Maximum Order

On my computer, I have tested the order. It has shown to work until 1449. An order 1449 or above will trigger a segmentation error.