Computer Programming Language

【Fall, 2018】

Homework 5

Program A： Caesar Shift Cipher Decoder (20%)

Cryptography is one of most interesting branches of programming. Studying its algorithms usually begins with the simple method named after famous Roman emperor Julius Caesar who used it for communicating his military secrets. The idea of the Caesar shift cipher algorithm is simple. Each letter of the original text is substituted by another, by the following rule:

(1) find the letter (which should be encrypted) in the alphabet;
(2) move K positions further (down the alphabet);
(3) take the new letter from here;
(4) if "shifting" encountered the end of the algorithm, continue from its start.

For example, if K = 3, then A becomes D, B becomes E, W becomes Z and Z becomes C and so on.

Your task is to write a Caesar shift cipher decoder program to decode the following encrypted sentence given K = 6:

O RUBK IUSVAZKX VXUMXGSSOTM YU SAIN

■ *Web-Cat Submission Check:*

```
char answer1;   // Store the first character of the decoded sentence
char answer2;   // Store the last character of the decoded sentence
```

Optional Bonus Points (20%):

You are encouraged to challenge a more difficult problem by designing a Caesar shift cipher cracker program to automatically find out the encrypted sentence without knowing the value of K. The encrypted sentence is as follows. Read the Wikipedia article about Caesar shift cipher algorithm to learn more about the approach to crack the encrypted sentence (https://en.wikipedia.org/wiki/Caesar_cipher).

GWC IZM ZMITTG I PIZL EWZSQVO ABCLMVB

■ *Web-Cat Submission Check:*

```
int answer1;    // The value of K to decode the above encrypted message.
char answer2;   // Store the first character of the decoded sentence
char answer3;   // Store the last character of the decoded sentence
```

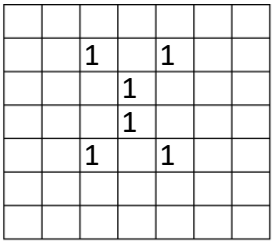
Program B： Game of life (40%)

The game of life is a computer simulation that was created by a Cambridge Mathematician named John Conway. The idea is that in each generation life will populate, survive, or die based on certain rules. Read the Wikipedia article (http://en.wikipedia.org/wiki/Conway's_Game_of_Life) to learn more about this famous simulation.

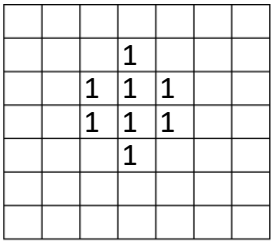
The universe of the Game of Life is a two-dimensional orthogonal grid of square cells, each of which is in one of two possible states, alive or dead. Every cell interacts with its eight neighbors, the immediately adjacent cells in orthogonal and diagonal direction. Cells on the border have less than eight neighbors. At each generation, the following transitions occur:

1. Any live cell with fewer than two live neighbors dies, as if by loneliness.
2. Any live cell with more than three live neighbors dies, as if by overcrowding.
3. Any live cell with two or three live neighbors lives, unchanged, to the next generation.
4. Any dead cell with exactly three live neighbors comes to life.

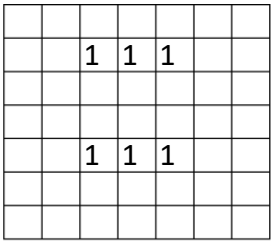
The following figures show the grid examples for the first generation, second generation, and third generation, respectively.



First generation



Second generation



Third generation

Your task is to write a program to simulate the Game of Life for a 20x20 world. The 20x20 world needs to be initialized by reading in which cells are occupied from the user. Your program will generate and display the next generation iteratively based on the set of rules described above. The number of generation for the simulation is also input by the user.

■ *Web-Cat Submission Check:*

```
int answer1;    // Store the total number of live cells of the first generation
int answer2;    // Store the total number of live cells of the second generation
int answer3;    // Store the total number of live cells of the third generation
```

Program C： Magic square (40%)

A magic square is a square of numbers with *N* rows and *N* columns, in which each integer value from 1 to (*N*×*N*) appears exactly once, and the sum of each column, each row, and each diagonal is the same value. For example, the following figure shows a magic square in which *N* = 3, and the sum of the rows, columns, and diagonal is 15. Write a program that constructs and displays a magic square for any given odd number N. This is the algorithm:

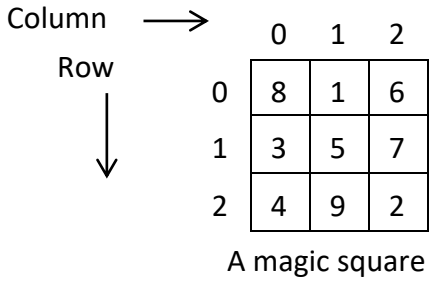
Insert the value 1 in the middle of the first row (element [0][N%2]).
After a value, x, has been placed, move up one row and to the right one column.
Place the next number, x+1, there, unless:

(5) You move off the top (row = -1) in any column. Then move to the bottom row and place the next number, x+1, in the bottom row of that column.

(6) You move off the right end (column = N) of a row. Then place the next number, x+1, in the first column of that row.

(7) You move to a position that is already filled or out of the upper-right corner. Then place the next number, x+1, immediately below x.

Stop when you have placed as many elements as there are in the array.



A magic square

■ *Web-Cat Submission Check:*

```
int answer1;    // Store the integer value of the cell at the top-left corner
int answer2;    // Store the integer value of the cell at the bottom-right corner
```

Notes:

1. Please submit your programs (source codes) to the Web-CAT grading system website (http://140.112.94.129:8080/Web-CAT_1.4.0/WebObjects/Web-CAT.woa/) before **Nov. 29**. (3:30PM)

2. Late submission will have a penalty of 10% discount per day of your grade toward a minimum score of 60. No late submission over a week will be accepted.

3. Criteria of grading include: (1) Program functionality; (2). User interface; (3). Structure of the program; (4). Suitable comments; (5). Programming style; (6). Creativity. The proper use of functions in building a modular program is encouraged and will be part of the grading criteria in this homework.