Creative Software Programming, Assignment 4-2

Handed out: Sep 30, 2021

Due: 23:59, Oct 5, 2021 **(NO SCORE for late submissions!)**

- *Only files submitted by **git push to this course project at** https://hconnect.hanyang.ac.kr (<Year>_<Course no.>_<Class code>/<Year>_<Course no.>_<Student ID>.git) will be scored.*
- *Place your files under the directory structure **<Assignment name>/<Problem no.>/<your files>** just like the following example.*

```
+ 2020_ITE0000_2019000001
  + 2-1/
    + 1/
      - 1.cpp
      - Makefile
    + 2/
      - 2.cpp
      - Makefile
    + …
```

- *The submission time is determined not when the commit is made **but when the git push is made**.*
- *Your files must be committed to the **master branch**. Otherwise, it will not be scored.*
- *Your program should output correct results even for inputs other than those used in the example.*
- *Basically, assignments are scored based on the output results. If it is not possible to check whether a requirement is implemented because the output is not correct, no score is given for the requirement, even if it is implemented internally. However, even if the output result is correct, no score is given for a requirement if the internal implementation does not satisfy the requirement.*

1. Write a program that creates a "magic square" of odd order.

A. A magic square is a n × n square grid filled with distinct positive integers in the range 1 , 2 , . . . , n^2 such that each cell contains a different integer and the sum of the integers in each row, column and diagonal is equal [wikipedia].

B. How to create a magic square of odd order:

   i. https://en.wikipedia.org/wiki/Magic_square#A_method_for_constructing_a_magic_square_of_odd_order

C. Take an integer N from the user.

   i. If N is not an odd number greater than or equal to 3, just exit your program.

D. The magicSquare() function should take an n x n matrix or (n * n) array and fills each element with the value of the magic square.

E. Print out the magic square in the main()

F. Note that

   i. An array (or matrix) to pass to the magicSquare() function must be dynamically allocated.

   ii. DO NOT include <stdio.h>, <cstdio>.

   iii. DO NOT use malloc() and free().

G. Input: One odd number greater than or equal to 3

H. Output: The magic square of the given size N

I. Files to submit:

   i. A C++ source file

   ii. A Makefile to generate the executable

```
$ ./magic_square
3
8 1 6
3 5 7
4 9 2
$ ./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
$
```

2. Write a program that creates a “magic square” of even order.

   A. All conditions are the same except that the user input is an even number.

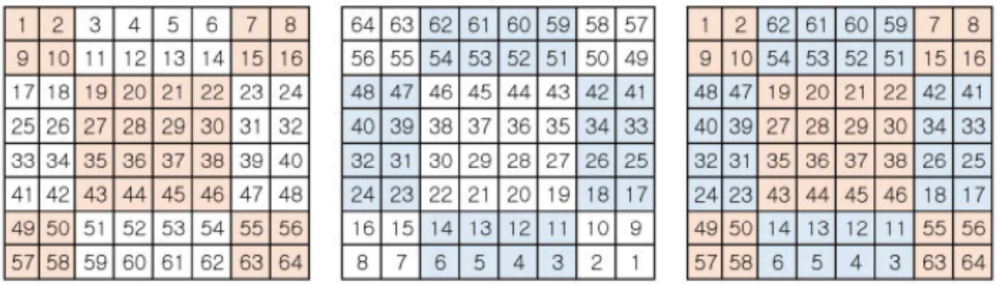
      i. Input even numbers will be given as **multiples of 4.**

   B. Divide the input number in ratio 1:2:1 (ex. 8 = 2+4+2).

   C. Fill in a matrix in ascending order.

   D. Fill in another matrix in descending order.

   E. Combine the marked area.



1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

64	63	62	61	60	59	58	57
56	55	54	53	52	51	50	49
48	47	46	45	44	43	42	41
40	39	38	37	36	35	34	33
32	31	30	29	28	27	26	25
24	23	22	21	20	19	18	17
16	15	14	13	12	11	10	9
8	7	6	5	4	3	2	1

1	2	62	61	60	59	7	8
9	10	54	53	52	51	15	16
48	47	19	20	21	22	42	41
40	39	27	28	29	30	34	33
32	31	35	36	37	38	26	25
24	23	43	44	45	46	18	17
49	50	14	13	12	11	55	56
57	58	6	5	4	3	63	64

   F. The other methods are all valid only if the result is a magic square.

   G. Input: One even number, multiple of 4.

   H. Output: The magic square of the given size N

   I. Files to submit:

      i. A C++ source file

      ii. A Makefile to generate the executable

```
$ ./magic_square
4
16 2 3 13
5 11 10 8
9 7 6 12
4 14 15 1
$ ./magic_square
8
64 63 3 4 5 6 58 57
56 55 11 12 13 14 50 49
17 18 46 45 44 43 23 24
25 26 38 37 36 35 31 32
33 34 30 29 28 27 39 40
41 42 22 21 20 19 47 48
16 15 51 52 53 54 10 9
8 7 59 60 61 62 2 1
$
```