

# VG101 Lab Manual

## Lab 2

Instructor: Dr. Yifei ZHU

TA: Hangrui CAO

TA: Qinhang WU

TA: Muchen XU

## Table of Contents

- MISC: Github Tutorial
- Conditional Expression
- Loop
- Magic Square

## Workflow

Content	Approx. Time
Github Tutorial	15 mins
Warmup: Basic Syntax Exercise	30 mins
Break	10 mins
Problem: Magic Square	45 mins
Break	10 mins
Practical Exercise	70 mins

## GitHub



[GitHub](#) is a community for developers to share their codes. Many wonderful open-source projects like [Linux](#), [Ruby](#), [Swift](#), and [PyTorch](#), are available on GitHub.

# Basic Concepts in GitHub

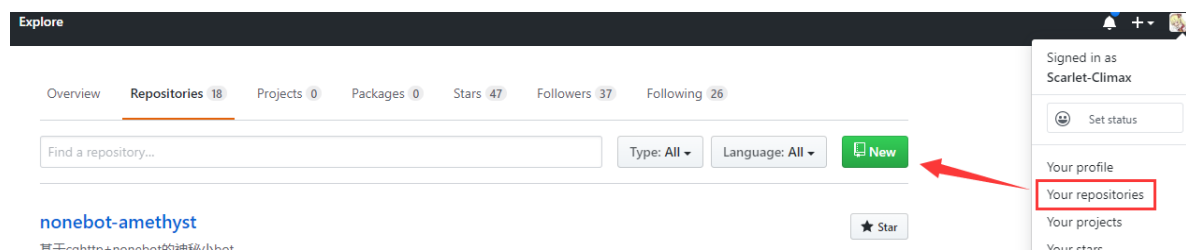
- **Repository**
  - where the code is stored, like a cloud drive.
  - can be considered as a project.
- **Commit**
  - the action of uploading (updating) the code to the repository.
  - measures how dedicated a programmer is.
- **Clone**
  - the action of downloading the code to the computer.
- **Git**
  - a software related to GitHub, **but is not GitHub**.
  - aims at project version control, i.e. easily recover your project when you mess up
  - will discuss next week.
- **Pull Request**
  - how people collaborate on GitHub.
  - improve other's project by modifying their code
  - need approval
- **Fork**
  - the action of "cloning" other's repository for your own use.

Before looking into the GitHub, please create an account in the GitHub.

## Use GitHub Online

### Create a Repository

Repository is where you store an entire project.



Name your repository and choose privacy.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner

Scarlet-Climax ▾

/

Repository name \*

Test ✓

Great repository names are short and memorable. Need inspiration? How about [expert-guide?](#)

Description (optional)

test

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None ▾

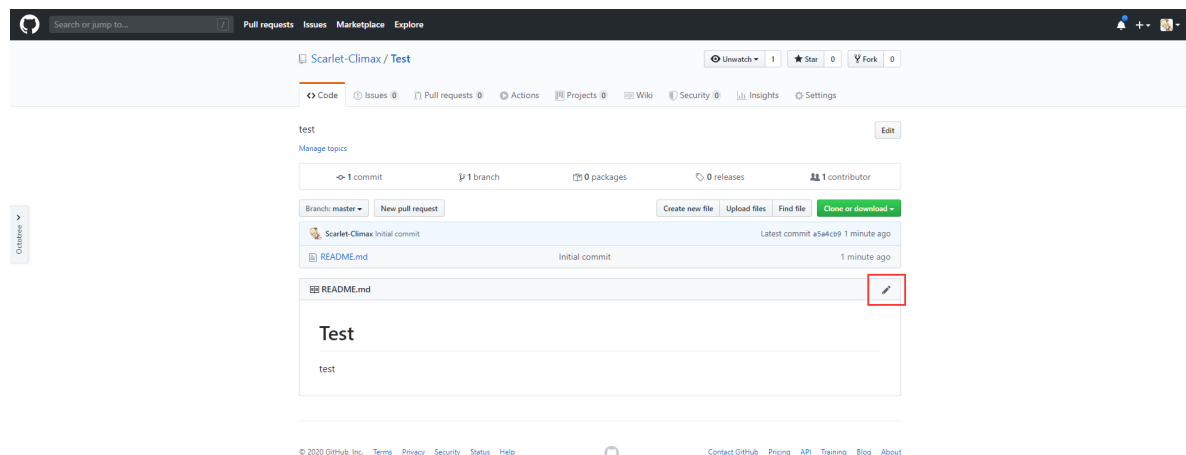
Add a license: None ▾

[i](#)

Create repository

## Commit the First Change

Interface of a repository:



Committing a change of the `README.md` will also change the appearance in the repo interface.

## Exploring GitHub

[GithubExplore](#) will give you information you may interested in:

1. Topics
2. Trending
3. Collections
4. Events
5. GitHub Sponsors

## Optional Assignment

Create a repository of your own and write a brief description of one of the repositories you are interested in on the GitHub, into the `README.md` of you repo.

You are supposed to use features in Markdown including `header` and `lists`.

Submit **the URL of your repo** (where you can directly see the rendered README file) in the assignment.

## Conditional Expression for MATLAB

### Josephus Game

In the Jewish revolt against Rome, Josephus and 39 of his comrades were holding out against the Romans in a cave. With defeat imminent, they resolved that, like the rebels at Masada, they would rather die than be slaves to the Romans. They decided to arrange themselves in a circle. One man was designated as number one, and they proceeded clockwise killing every seventh man... Josephus (according to the story) was among other things an accomplished mathematician; so he instantly figured out where he ought to sit in order to be the last to go. But when the time came, instead of killing himself he joined the Roman side.

In our Josephus game, we start with  $n$  people numbered 1 to  $n$  around a circle, and we eliminated every  $k$  remaining person until only one survives. For example, here's the starting configuration for  $n = 10$ ,  $k = 2$ , The elimination order is 2, 4, 6, 8, 10, 3, 7, 1, 9. So 5 survives.

Please determine the survivor's number for input  $n$  and  $k$ .

```
1  >>ex12
2  10
3  2
4      5
```

#### Hint

In MATLAB, it is simple to delete one element. One way is to use the double loop.

### Magic Square

A magic square is a  $n \times n$  square grid (where  $n$  is the number of cells on each side) filled with distinct positive integers in the range  $1, 2, \dots, n^2$  such that each cell contains a different integer and the sum of the integers in each row, column and diagonal is equal. The sum is called the magic constant or magic sum of the magic square. A square grid with  $n$  cells on each side is said to have "order  $n$ ".



step 1			step 2			step 3			step 4		
	1			1			1			1	
						3			3		
					2			2	4		2

step 5			step 6			step 7			step 8			step 9		
	1			1	6		1	6	8	1	6	8	1	6
3	5		3	5		3	5	7	3	5	7	3	5	7
4		2	4		2	4		2	4		2	4	9	2

## Magic Square of Doubly Even Order

In this part, you're required to write a MATLAB script which takes in an **doubly even** integer  $n > 2$  as input, and return a magic square of order  $n$  as a matrix similar to MATLAB function `magic(n)`.

Recall that doubly even means that  $n$  is an even multiple of an even integer, i.e.,  $4, 8, 12, \dots$ , where  $k$  is an integer. Here is an approach which is called *Narayana-De la Hire's method*.

All the numbers are written in order from left to right across each row in turn, starting from the top left hand corner. The square is split into  $n^2/16$  parts of  $4 \times 4$  small squares. Numbers in the small squares are then retained in the same place if they are not on the diagonal line, or interchanged with their diametrically opposite numbers if they are on the diagonal line. Two numbers  $a, b$  are diametrically opposite means that  $a + b = n^2 + 1$ .

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

## Magic Square of Singly Even Order

We will not cover this part in our lab. If interested, you can search for `Conway's LUX method` by yourself.

## Print a Magic Square

In this part, you're required to print a generated magic square of odd order in a `.txt` file. The workflow is listed as follow:

- Read an integer  $n$  from standard I/O.
- Open a file `output.txt` with write permission.
- Call the function you wrote in the previous section with the integer  $n$  that you input before and print the returned matrix in `output.txt`. If the integer doesn't meet your function (i.e.,  $n = 4$ ), print some error message.

### Remarks

- Don't forget to close the opened file before your script ends.
- You're recommended to use `monospaced font` in MATLAB (and later, in other coding environments). Otherwise, you'll find it hard to align the numbers!

## Grading Rubric

Criteria	Weight	Available Time	Due Time	Entry
Attendance	30%	4:00pm, May.22	11:59pm, May.22	Canvas Assignment
In-lab quiz	70%	9:00pm, May.22	11:59pm, May.22	Canvas Quiz
Github bonus	10%	9:00pm, May.22	11:59pm, May.22	Canvas Assignment
Challenges bonus		9:00pm, May.22	2:00pm, May.23	Canvas Quiz

- For the attendance score, you need to submit your code for **exercises** in the **worksheet** on Canvas. We won't judge the correctness of your code. You'll earn full credits as long as you've tried these exercises.
- The attendance, in-lab quiz, and the Github bonus will be ceiled together if greater than 100%.
- The challenges bonus will be counted towards your final lab score.

## Reference

- VG101 Lab Manual (Summer 2019)