# C Programming II
# 2022 Spring
# Final

Instructor: Po-Wen Chi

Date: 2022.06.04 PM 14:00-18:00

**Policies**:

- Offline test unless you complete the exam and want to submit your code.
- Do not forget to include your Makefile. TA will only use the command make to build your program. If make fails, you will get zero points and no room for bargaining. So if you do not know how to solve a problem, please, do not include it in your Makefile.
- I do not care your source code file names, but the executive binary names should be **mid01, mid02, mid03, mid04**.
- You can ask TA if you do not understand the problems.

## 1 Color Gradient (35 pts)

In computer graphics, a color gradient specifies a range of position-dependent colors, usually used to fill a region. Now I want you to write a program to generate a BMP file to show the color gradient, like figure 1.

```
$ ./fin01 --help
Usage:
  ./fin01 [options]
    -w: the width of the output figure. (Default: 1024)
    -h: the height of the output figure. (Default: 768)
    -o: the output file name. (Default: output.bmp)
    --help: this message
$ ./fin01
Please enter (R,G,B) in the top left pixel: 0,0,0
Please enter (R,G,B) in the top right pixel: 255,0,0
Please enter (R,G,B) in the bottom left pixel: 0,255,0
Please enter (R,G,B) in the bottom right pixel: 0,0,255
Done
```

The colors of points are calculated using **linear interpolation**.

Figure 1: Color Gradient. This is **not** the result of the testing command.

## 2 COVID-19 (35 pts)

I want you to develop a program to plot the number of the daily new COVID-19 infected people of a country. First, you can get data from the following site.

`https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports`

You can download the raw data set there. You can also assume that libcurl is correctly installed on the TA's computer. You need to output a csv file to record the new infected number of each in this period. Moreover, you need to add one additional line to show the average. The new infected number of a day is defined as the confirmed number minus the previous day's confirmed number.

```
$ ./fin02 -c Taiwan -s 01-02-2020 -e 12-31-2021 -o output.csv
$ ./fin02 --help
fin02:
    -c, --country: the country name. default: Taiwan
    -s, --start MM-DD-YYYY: the start date. default: 01-02-2020.
    -e, --end MM-DD-YYYY: the end date. default: 12-31-2021.
    -o, --output: output file name. default: output.csv
    --help: This description
$ cat output.csv
Date, New infected
01-02-2020,15
01-03-2020,17
...
```

```
12-31-2021,12
Average,7
// The numbers are random and may not be correct!!
```

Notes:

- To download the data, you can check the real URL from the RAW button as shown in Fig. 2.
- Note that you need to check Taiwan in the dataset since its name is "Taiwan*". As for other countries, I promise the testing country name is definitely the same with the data.
- Some countries may have multiple regions. Do not forget to sum up all regions.
- I promise the given data is with valid format and there is a dataset of the date.

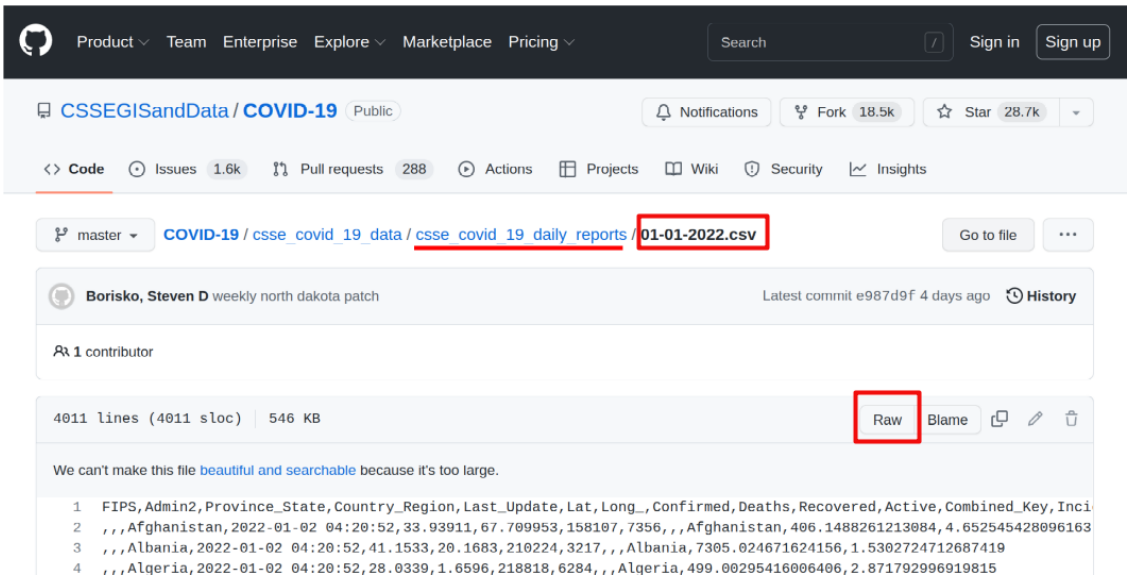


Figure 2: COVID-19 Data.

# 3 Maze (30 pts)

A maze is a path or collection of paths, typically from an entrance to a goal. Now, give you a maze, please find a **shortest path** from the entrance and the exit. Now I will show how you will get the maze.

A maze is composed of many nodes and many paths between these nodes. The node structure is as follows:

```
#include <stdint.h>
#include "linuxlist.h"

typedef struct _sNode
```

```
{
  int32_t id;
  int32_t flag;    // 0: entrance; 1: exit; 2: otherwise
  char    name[32]; // Location name.

  struct list_head next_node_list; // sNodeAddress List
} sNode;

typedef struct _sNodeAddress
{
  sNode *pNode;

  struct list_head list;
} sNodeAddress;
```

**next_node_list** records every node that the node has a path to it. For your simplicity, the path is **unidirectional**. That is, $A \rightarrow B$ does not imply $B \rightarrow A$. You need to include **maze.h** in your code and call **maze_gen()** at the start of your main function. You will get **sNode *** as the starting point. I will give you **maze.h** and **maze.o** will be prepared by our TAs. The output should look like this.

```
$ ./fin03
1(Start) -> 2(Basement) -> 5(Armory) -> 8(Throne) -> 6(Exit)
```

If there are multiple shortest paths, for your simplicity, you only need to print one path.

# 4 Bonus: Your Comments (5 pts)

Again, any comments are welcomed. However, you will get nothing if you leave this question blank.

# 5 Bonus: How do you take the Youtube Course? (5 pts)

In this semester, most students study programming from Youtube. I want to know when and where you see videos since most of you do not follow the live streaming. Will you prefer Google Meet or other video conference platforms instead of Youtube? Why or Why Not?