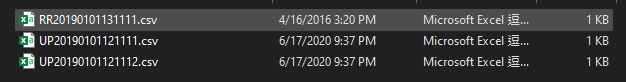
**197020323 Lau Ming**

**Proper implementation of progress control**

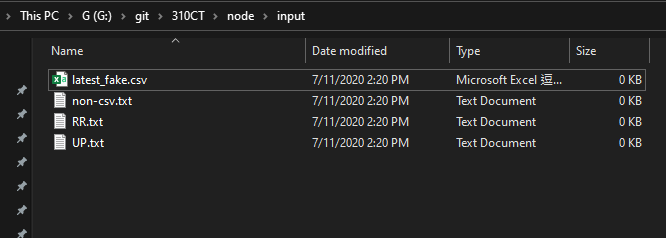
The efficiency, sequence and file movement have been handled properly.

This application will group all csv begin with (RR or UP) and check file format(csv)

Then, it will handle UP csv, then RR csv



These types of data would not be handled



**Proper priority control of file uploading and model updating**

It will handle UP csv, then update the current model

Then, handle the RR csv (using the latest model that already handled the up csv)

Then, it will response the items id.

**Proper implementation of recommendation logic**

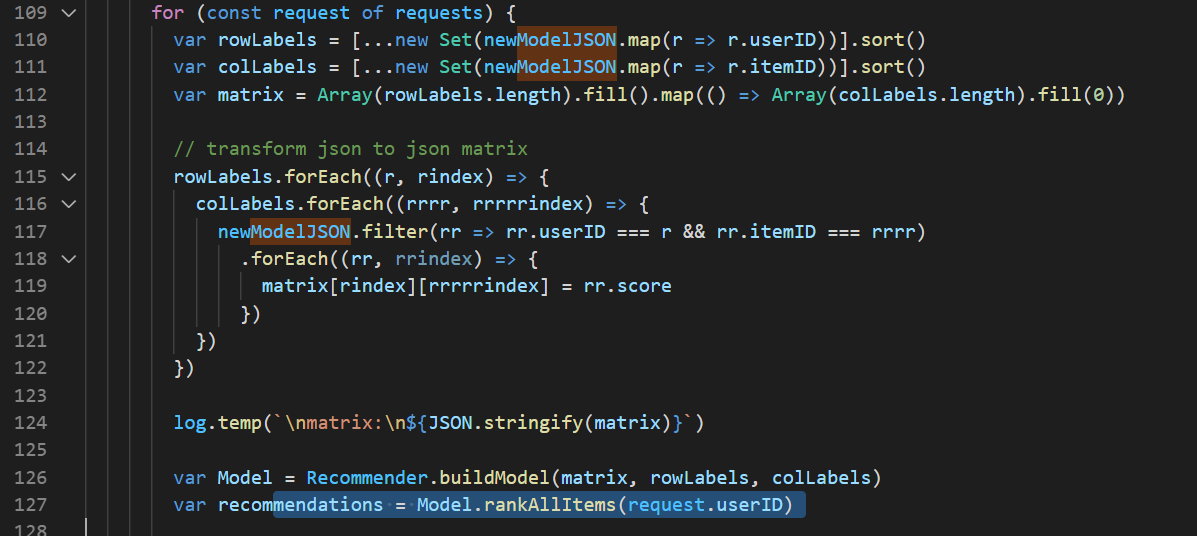
Using javascript library for collaborative filtering and recommendation engines called ¡¥likely¡¦

https://www.npmjs.com/package/likely

It needs to convert csv to json to json matrix to generate the result, cannot use the csv file directly

**Proper recommendation logic implemented**

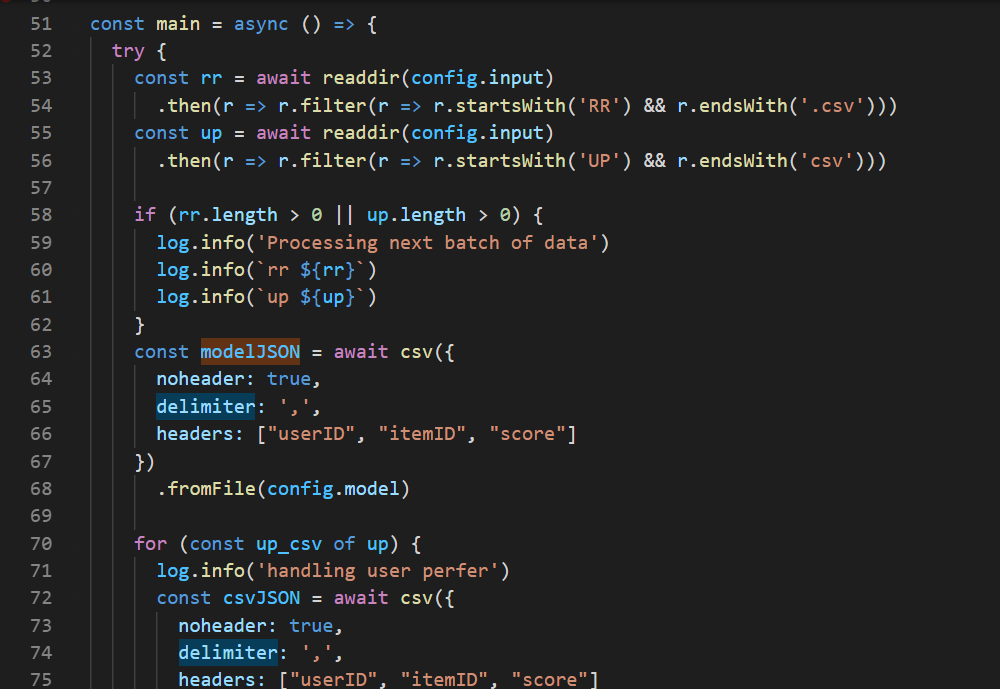
It will get the recommendations by request user id (line 127),

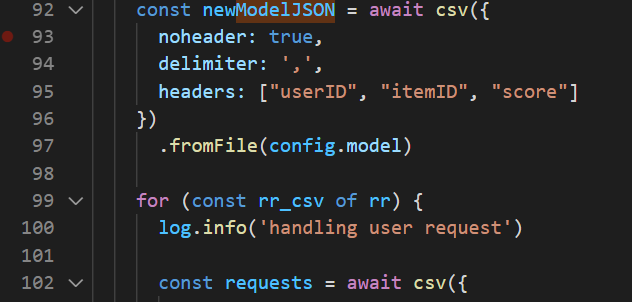


**Different scenarios can be supported with a reasonable response time.**

Result will generate one by one, not every batch of data.

Some construction data defined out of the loop, so NodeJS will not defined in every loop





Less of the conditional checking and programming with async function to reduce the response time, also add concurrency in looping (for of loop), it supposes run every 4 data at the same time according to internet.

**Provision of design documentation to show**

***Please read the Read.md in the node folder***

This is the documentation with picture

How to start the program

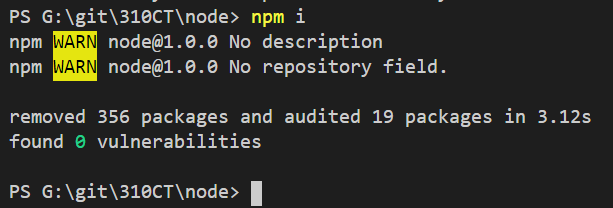
1. Unzip the file



1. Go into the node folder

`310CT\node`

1. run cmd `npm i` on terminal



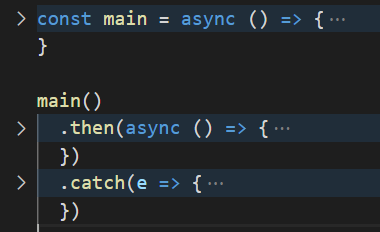
4. run cmd `node run.js`



The program will not stop until user stop it, so user can use cmd `node run.js > log.log &` to run program in background, user can tail log by the log.log file

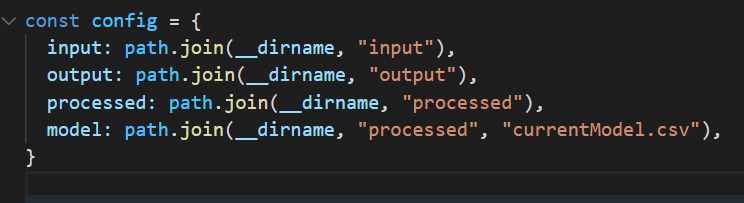
**b) Overall program flow**

When user start this program, NodeJS will call function `main`,

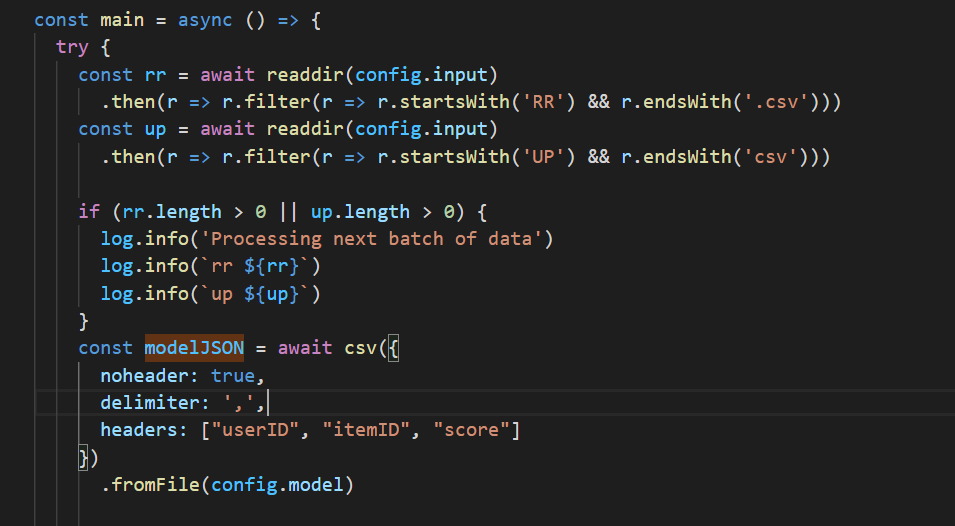


It will retry and log the error message, so this program will not stop if error happen

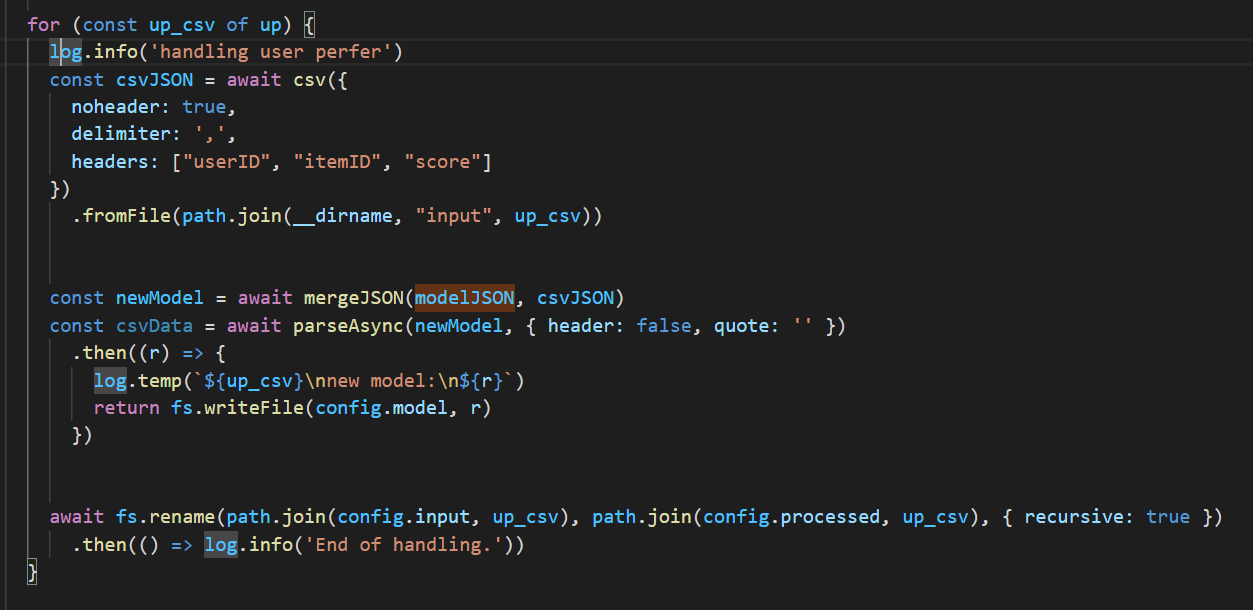
Then function main will read the directory which is delicate in config json

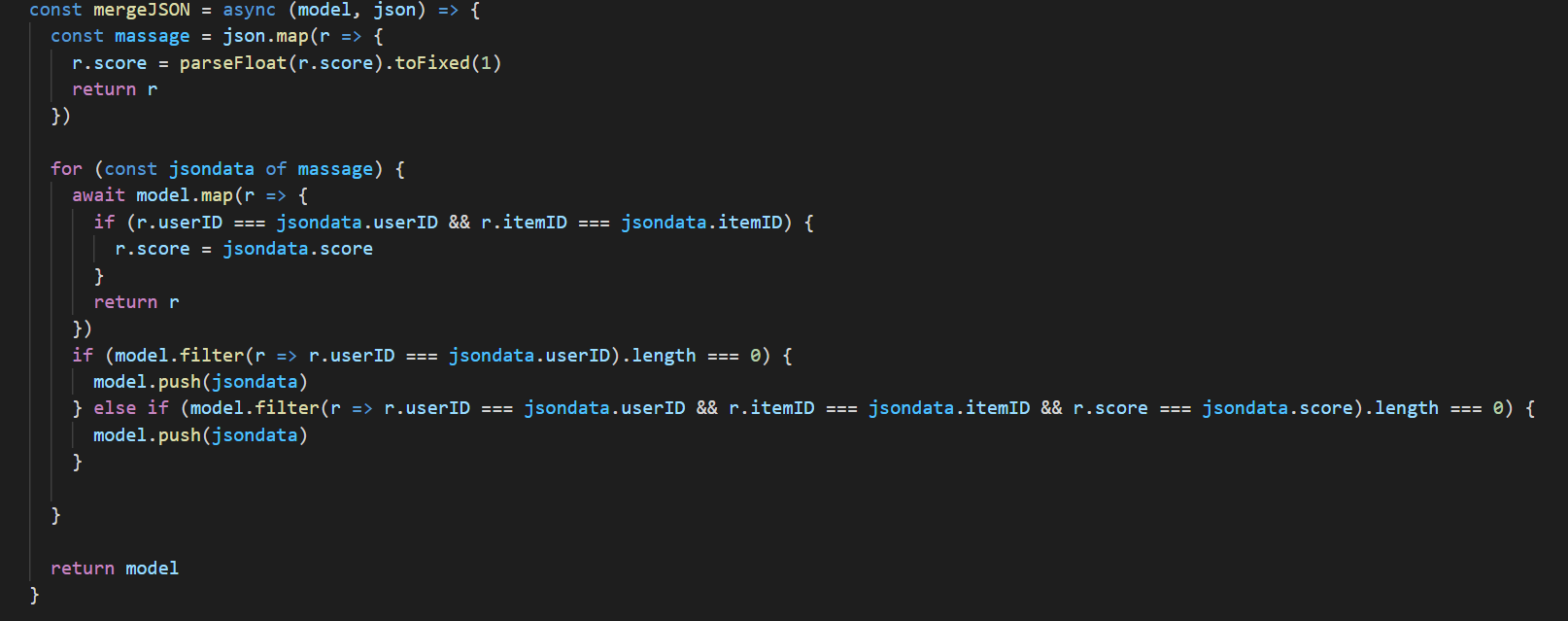


Then get the model csv before the loop, and covert to json and add the header for processing



Then, get the up csv and covert to json and merge to the model json and overwrite the current model, after that, move the up csv from input to processed folder

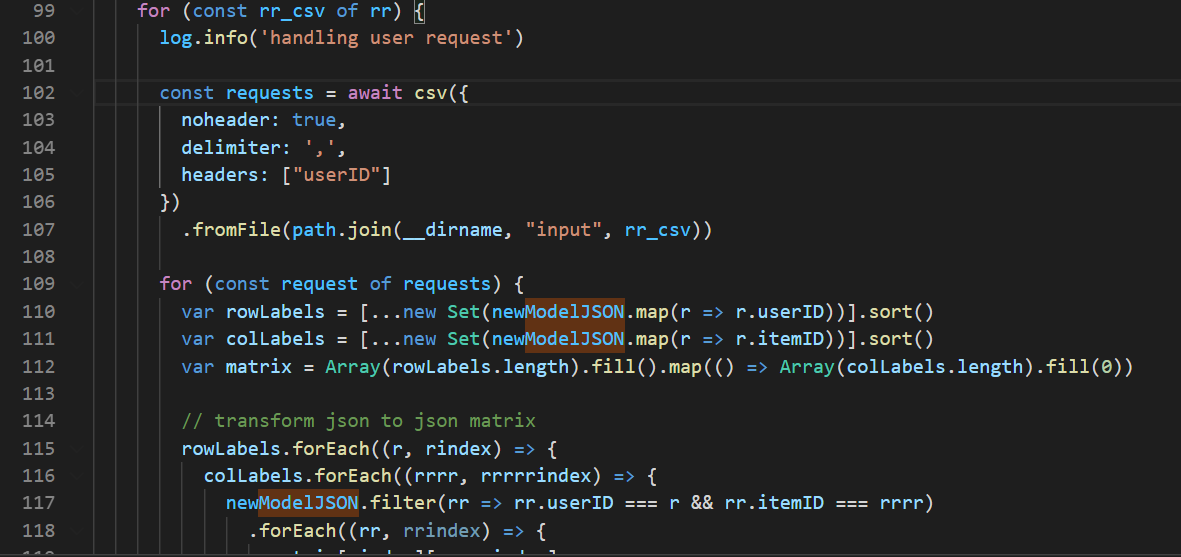




After handle all up csv, starting to process RR csv

It will read all RR csv and loop one by one, each rr.csv may have multi row, so there is another loop of each row of the RR csv.

Then, transform model csv to json matrix for library, the recommend result will write to the csv with the userID in the output folder. Then remove the request csv.

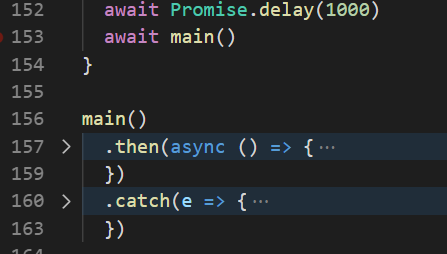


**Recommendation feature discussion including library discussion (strength and weakness)**

Strength:

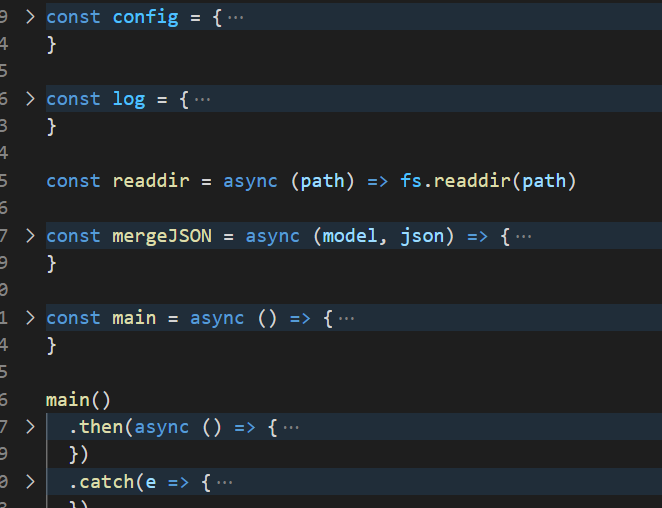
1. Retry/ Non-stop

User only need to start the program once, program will resume after 1 sec



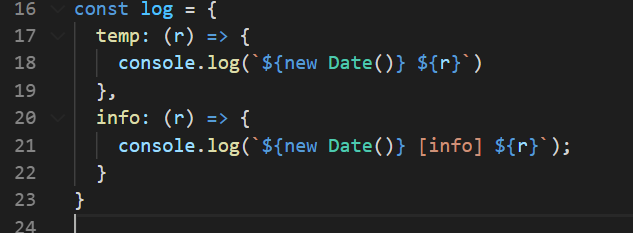
2. Functional Programming

Using different function instead of one big function



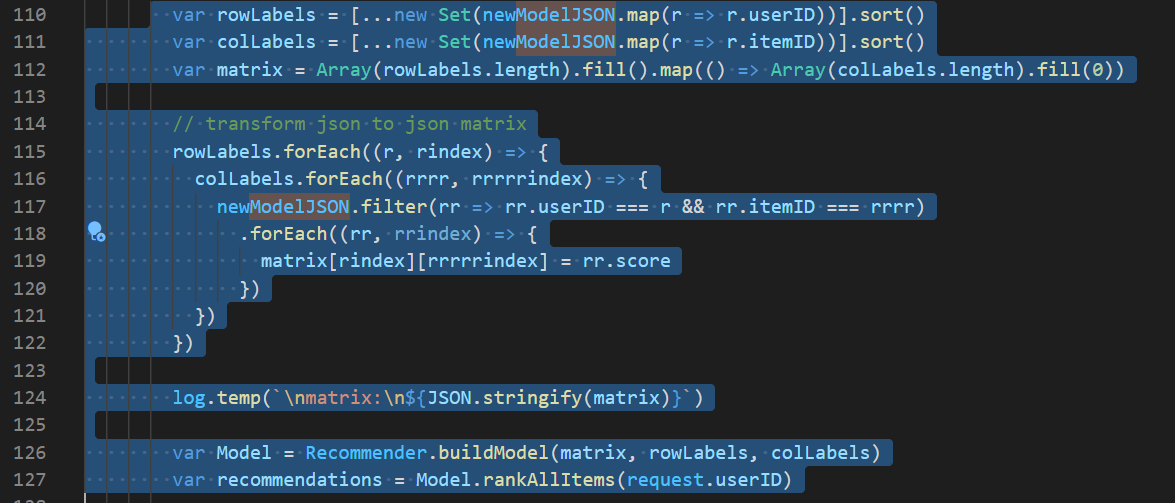
3. Log Message

The log message includes date and content because I make it functionality



4. library

this library needs to convert csv to json matrix, need more step and logic to handle data, if it can handle data directly is better.



Library is light weight, so install and resp very fast, also least dependence, so the version of library is not concern

Weakness:

Do not have any weakness so far

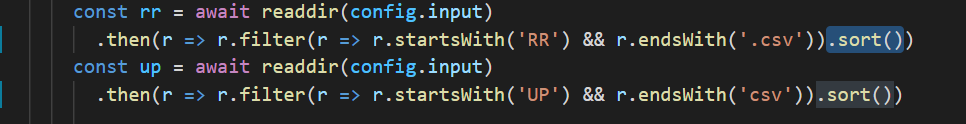
**Explanation of the program**

Overall:

Copy folder input data sample in G:\git\310CT\node\data sample\310CT that paste it to G:\git\310CT\node\

Program will process these data immediately, if any data are input during the processing, it will handle in next batch

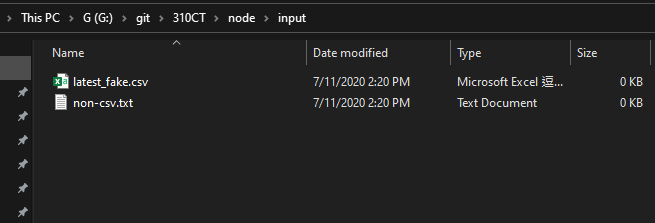
The SAME UP or RR csv will sort by the date(file name)



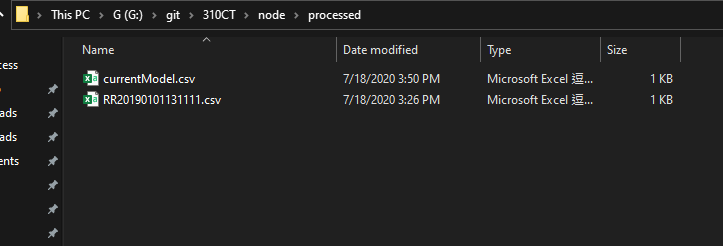
Details:

Before Start program:

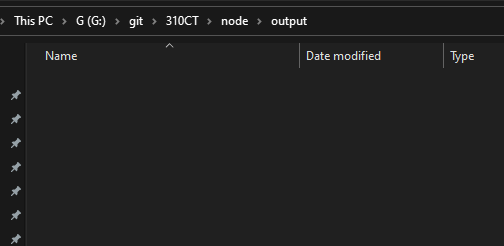
1. Input folder, having some fake data for testing (Program should handle these data)



2. Processed folder



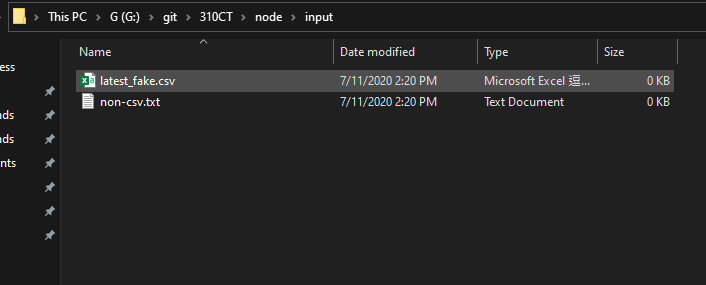
3. Output folder



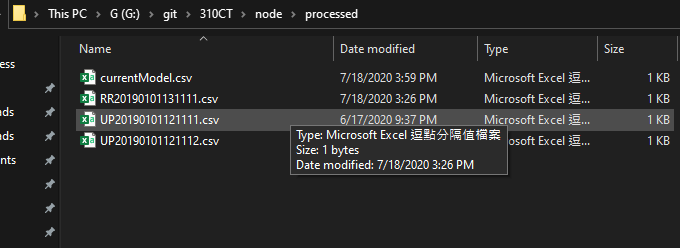
After starting the program

1. Insert csv to input folder

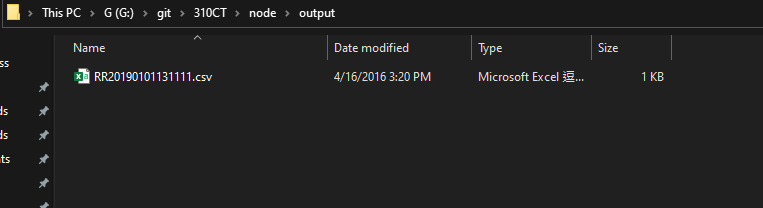
Input folder (csv will process immediately)

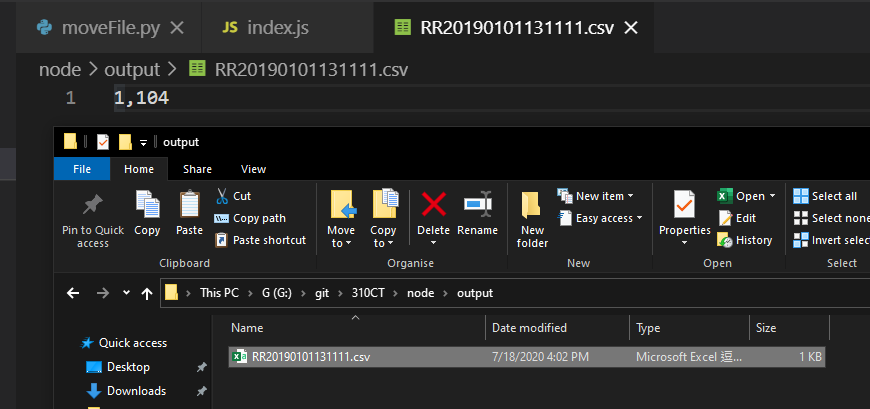


Processed folder



Output folder

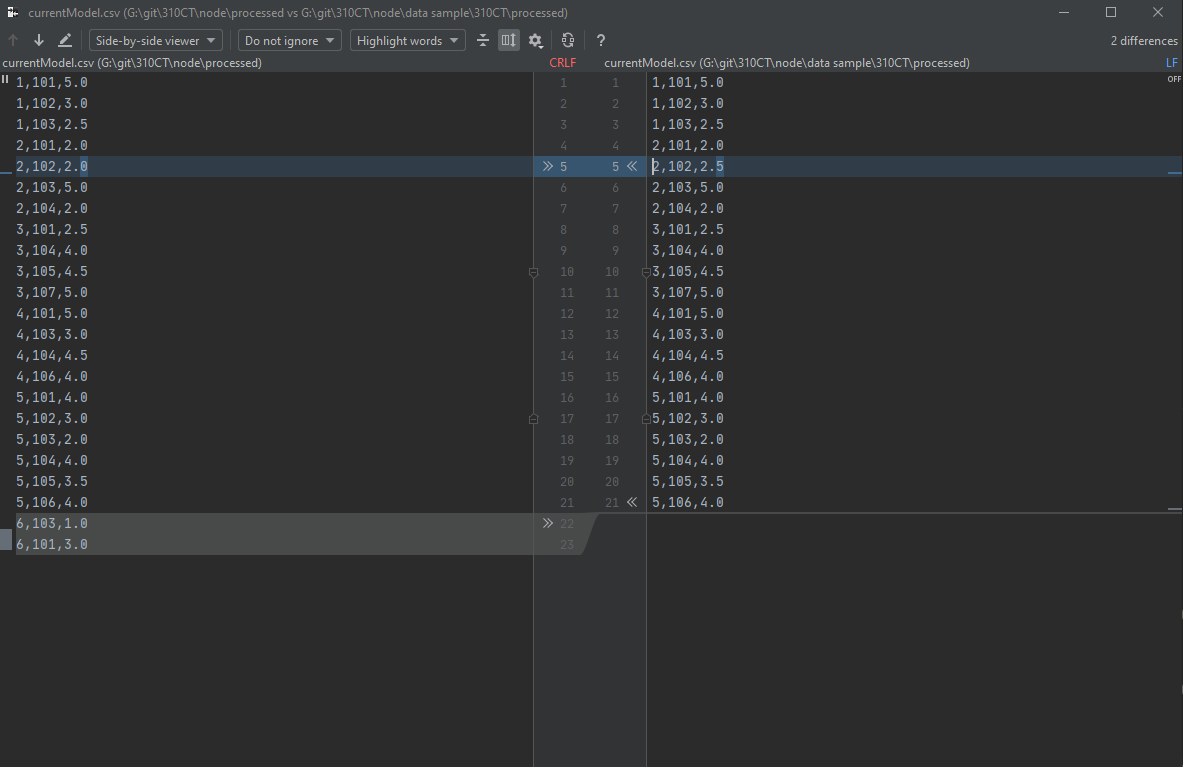




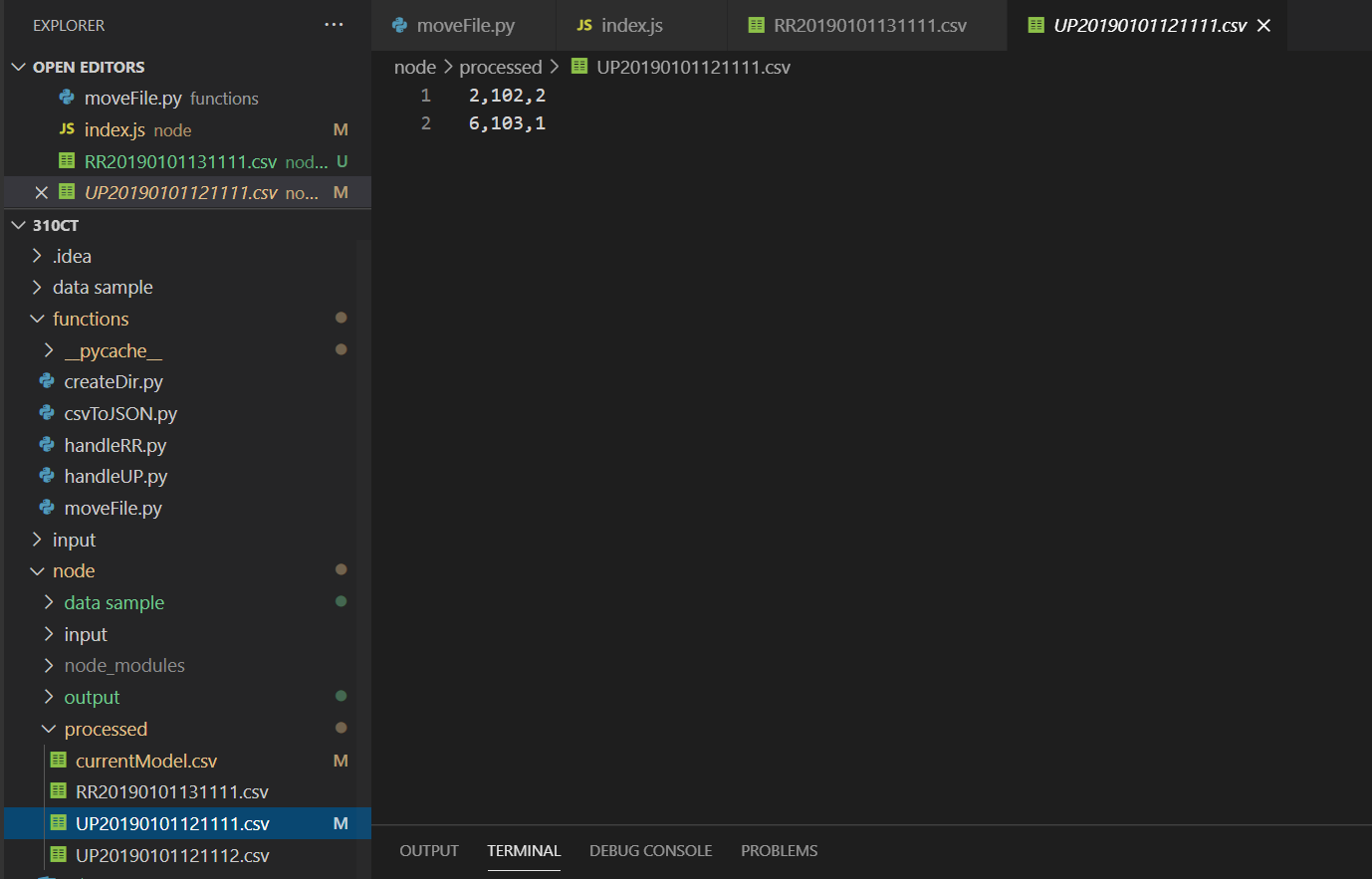
Test result examples

(Maximum 4 pages for a-c with font size 12)

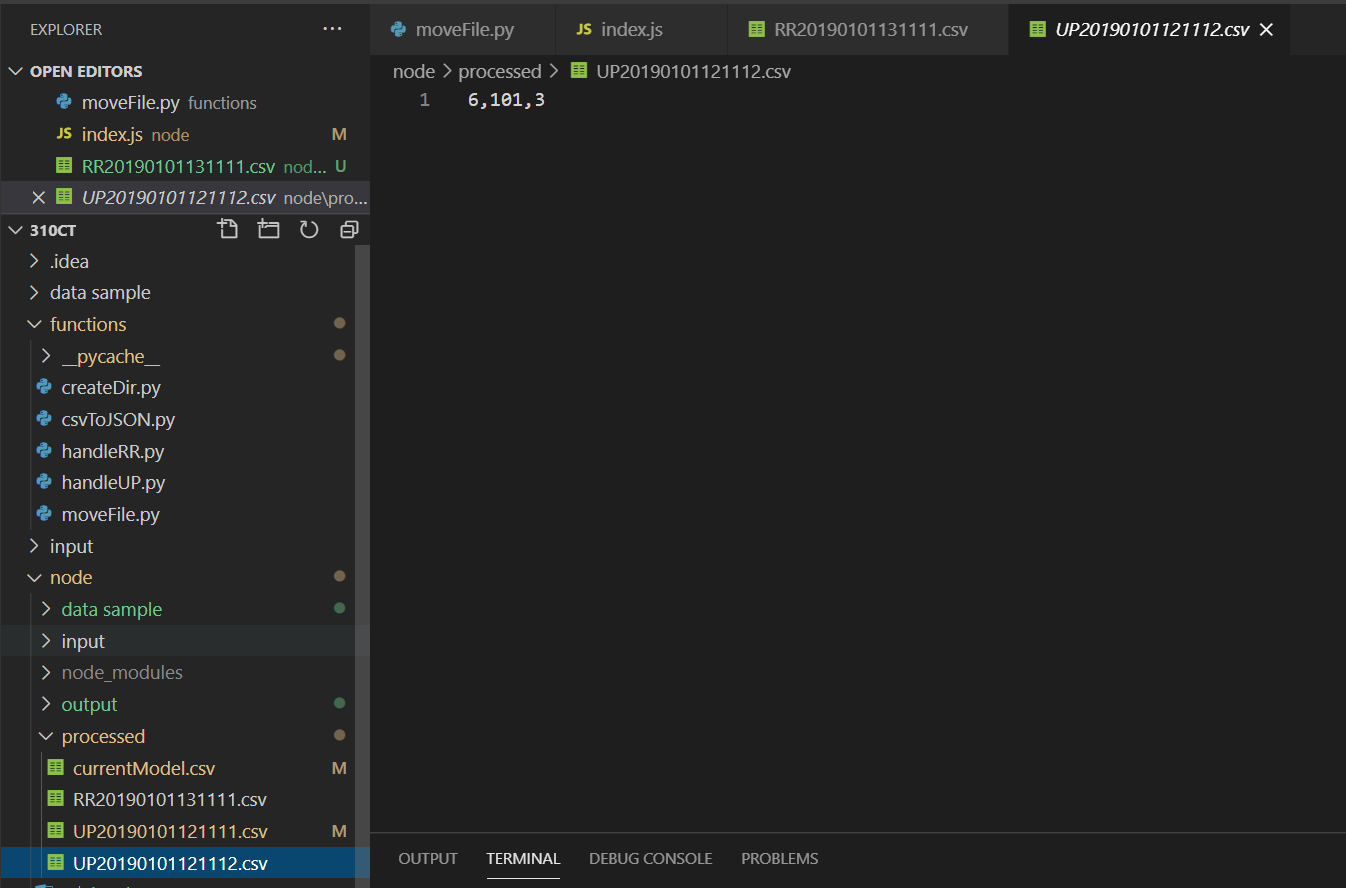
currentModel.csv After vs Before



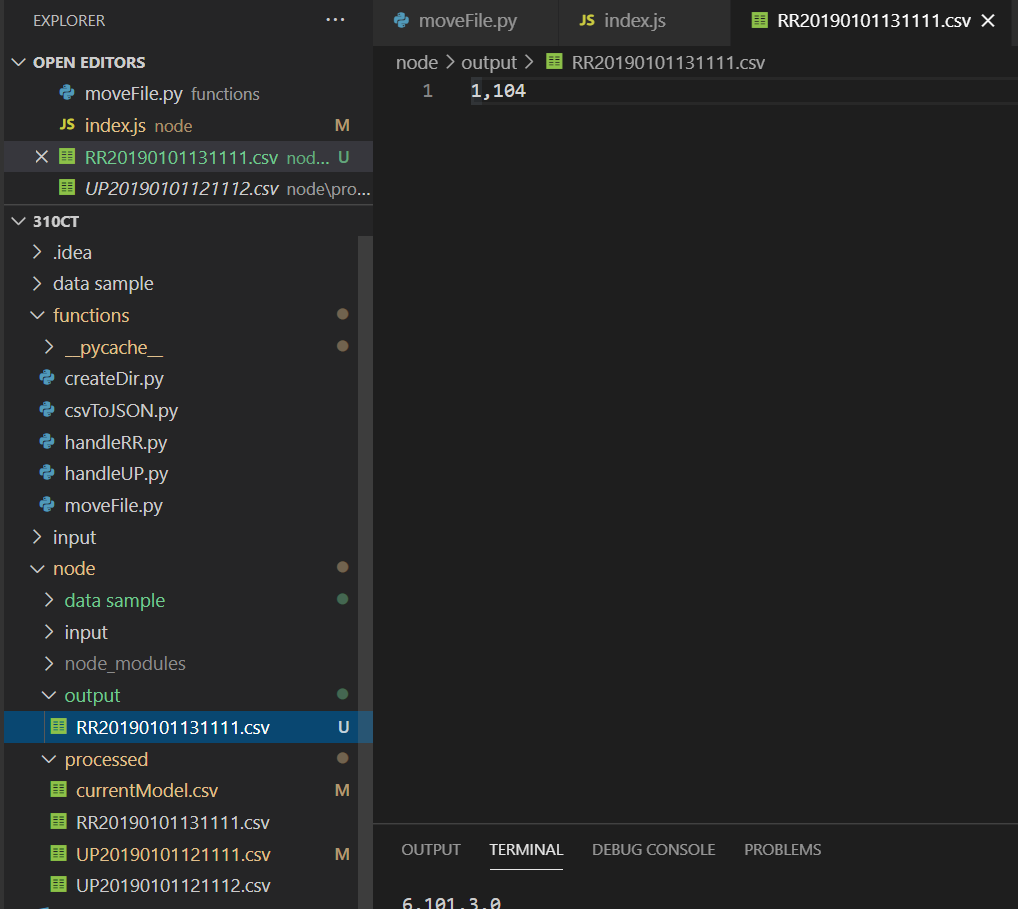
UP20190101121111.csv

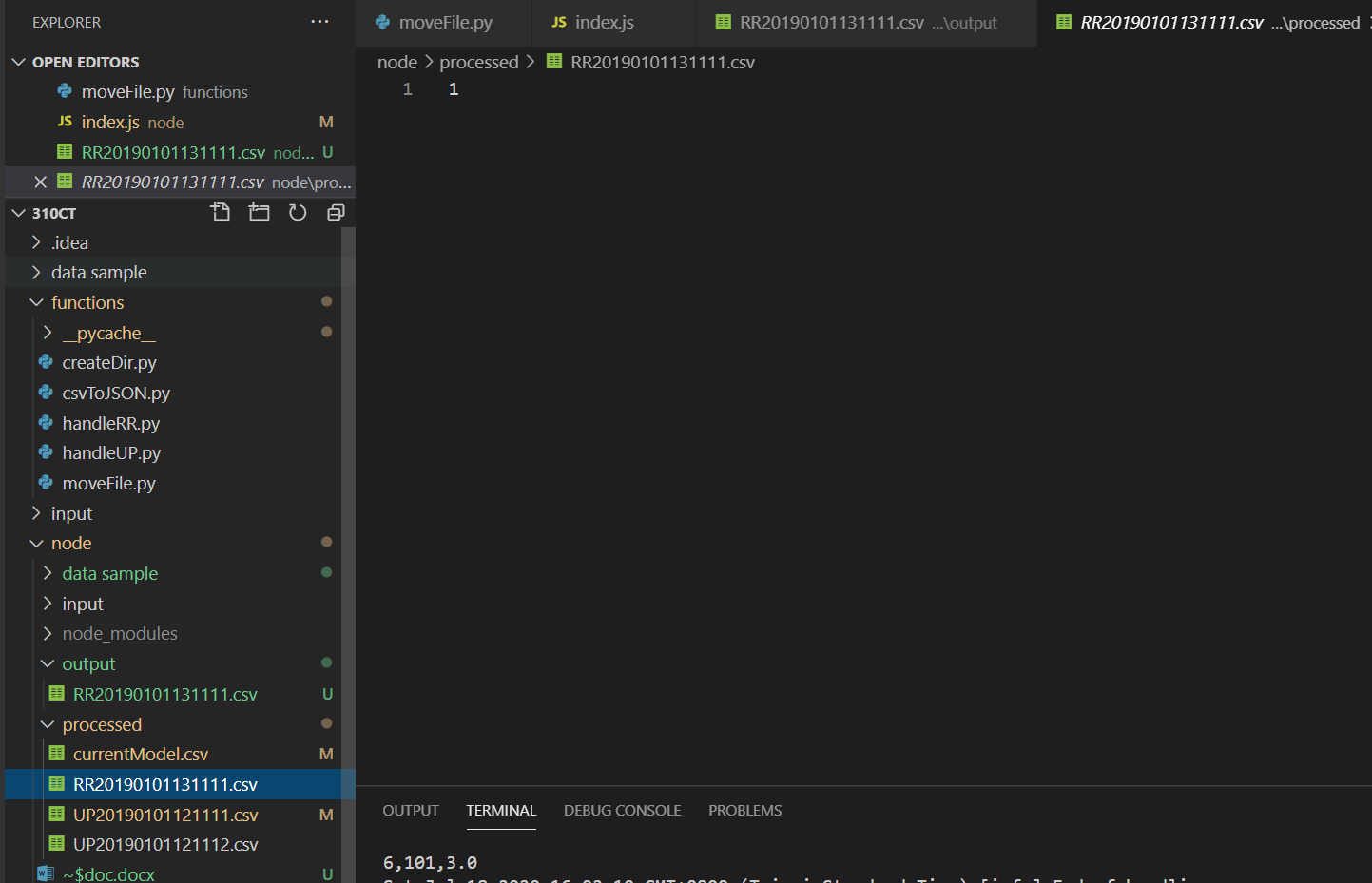


UP20190101121112.csv



RR20190101131111.csv





**LOG**

The log flows are easy to understand what happens and how to debug

|  |
| --- |
| PS G:\git\310CT\node> node .\index.js  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) [info] Processing next batch of data  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) [info] rr RR20190101131111.csv  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) [info] up UP20190101121111.csv,UP20190101121112.csv  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) [info] handling user perfer  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) UP20190101121111.csv  new model:  1,101,5.0  1,102,3.0  1,103,2.5  2,101,2.0  2,102,2.0  2,103,5.0  2,104,2.0  3,101,2.5  3,104,4.0  3,105,4.5  3,107,5.0  4,101,5.0  4,103,3.0  4,104,4.5  4,106,4.0  5,101,4.0  5,102,3.0  5,103,2.0  5,104,4.0  5,105,3.5  5,106,4.0  6,103,1.0  6,101,3.0  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) [info] End of handling.  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) [info] handling user perfer  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) UP20190101121112.csv  new model:  1,101,5.0  1,102,3.0  1,103,2.5  2,101,2.0  2,102,2.0  2,103,5.0  2,104,2.0  3,101,2.5  3,104,4.0  3,105,4.5  3,107,5.0  4,101,5.0  4,103,3.0  4,104,4.5  4,106,4.0  5,101,4.0  5,102,3.0  5,103,2.0  5,104,4.0  5,105,3.5  5,106,4.0  6,103,1.0  6,101,3.0  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) [info] End of handling.  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time) [info] handling user request  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time)  matrix:  [["5.0","3.0","2.5",0,0,0,0],["2.0","2.0","5.0","2.0",0,0,0],["2.5",0,0,"4.0","4.5",0,"5.0"],["5.0",0,"3.0","4.5",0,"4.0",0],["4.0","3.0","2.0","4.0","3.5","4.0",0],["3.0",0,"1.0",0,0,0,0]]  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time)  matrix:  [["5.0","3.0","2.5",0,0,0,0],["2.0","2.0","5.0","2.0",0,0,0],["2.5",0,0,"4.0","4.5",0,"5.0"],["5.0",0,"3.0","4.5",0,"4.0",0],["4.0","3.0","2.0","4.0","3.5","4.0",0],["3.0",0,"1.0",0,0,0,0]]  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time)  recommendations:  [["101",5.000424093605808],["102",3.0074650868278807],["103",2.4950023974185345],["104",4.3606836643990725],["105",3.616323036935864],["106",4.224056431439491],["107",4.1997105076099475]]  Sat Jul 18 2020 16:02:10 GMT+0800 (Taipei Standard Time)  Rank:4.3606836643990725  Item:104  Sat Jul 18 2020 16:02:11 GMT+0800 (Taipei Standard Time) [info] End of handling. |

**Version Control**

I am using GitHub as my VCS, I public this repo recently,

https://github.com/lauming1111/310CT

you can use `git clone https://github.com/lauming1111/310CT` to checkout my repo

or `wget [https://github.com/lauming1111/310CT`](https://github.com/lauming1111/310CT%60)

Also using the pull request to merge branch into master, this is good practices for version control

