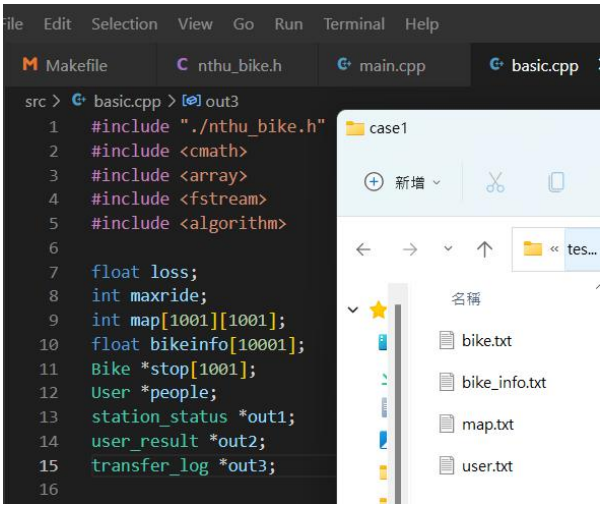
資料結構

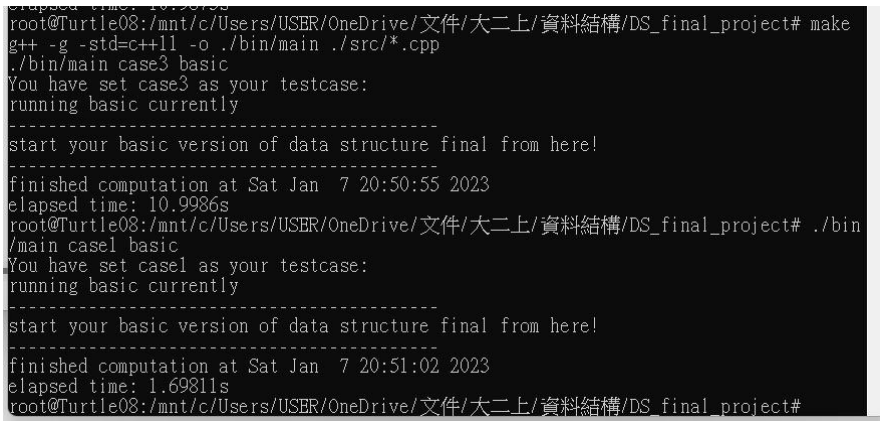
張育嘉 110030043

1. 執行

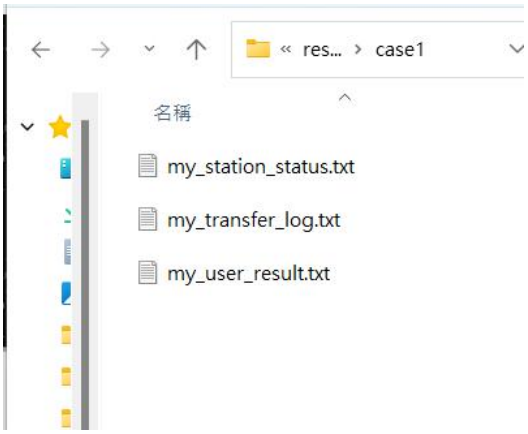
a) 先確認要 input 的四個檔案，和自己的 code(記得存檔)



b) 在 ubuntu 中 make 後執行並選擇你要執行的方式和檔案



c) 執行完在 result 中選擇要看的檔案



2.儲存資料

a) Map：用陣列儲存和計算資料(用 map 排序)

b) Bike: 儲存在 Node 中用 link list 串接起來(依照 rental price 再照 id 排序)

c) Bike_info: 儲存在全域陣列中

d) Stop: 用陣列代表每一個車站，同一個車站的車用 link list 串接起來，stop 是用來記錄當前時車站中有甚麼車在車站

```
float loss;
int maxride;
int map[1001][1001];
float bikeinfo[10001];
Bike *stop[1001];
User *people;
station_status *out1;
user_result *out2;
transfer_log *out3;
```

e) User: 儲存在 Node 中用 link list 串接起來(依照 starttime 再照 id 排序)

f) User_result: 儲存在 Node 中用 link list 串接起來(依照 userid 排序)

g) Transfer_log: 儲存在 Node 中用 link list 串接起來(依照 userid 排序)

```
8
9   typedef struct Node1
10  {
11      int type;
12      int id;
13      int oktime;
14      int befuserid;
15
16      float rentalprice;
17      int rentalcount;
18
19      Node1 *next;
20  } Bike;
21
22  typedef struct Node2
23  {
24      int id;
25      int starttime;
26      int endtime;
27      int startstop;
28      int endstop;
29      int biketype[10001] = {0};
30
31      Node2 *next;
32  } User;
```

```
src > nthu_bike.h > Node2
44
45  typedef struct Node4
46  {
47      int userid;
48      int isrent;
49      int bikeid;
50      int starttime;
51      int endtime;
52      int revenue;
53
54      Node4 *next;
55  } user_result;
56
57  typedef struct Node5
58  {
59      int bikeid;
60      int startstop;
61      int endstop;
62      int starttime;
63      int endtime;
64      int userid;
65
66      Node5 *next;
67  } transfer_log;
68
```

3.演算法

a) readmap: input map 檔案並用 Floyd-Warshall 的演算法來找出最短路徑

b) readbike: input bike 檔案並用排序資料(依照 rental price 再照 id 排序)

c) readstop: input stop 檔案並用排序資料(依照 station 再照 rential price 排序最後在照 bikeid 排序)

d) readbike_info: input bike_info 檔案並儲存在璇域變數區中的 loss, maxride, bikeinfo 裡面

e) basic 判斷區: 騎乘者(依照時間在照 userid 排序)是否會準時騎到站 -> 挑選腳踏車(type 是否符合 -> 現在是否可以把車借給他 -> rentalcount 是否超過 maxride) -> 可以借車

advance 判斷區: 騎乘者(依照時間在照 userid 排序)是否會準時騎到站 -> 挑選腳踏車

Case1 : 檢查 startstop 中的腳踏車 (type 是否符合 -> rentalcount 是否超過 maxride -> 移動車的時間合理嗎 和 結束時間是否會超過 endtime)

Case2 : 檢查 startstop 以外中的腳踏車 (type 是否符合 -> rentalcount 是否超過 maxride -> 移動車的時間合理嗎 和 結束時間是否會超過 endtime)

-> 可以借車

f) 如果租借成立: 新增 user_result 和 transfer_log -> 更新 bike 的資

訊

g) Insertstation_status: 儲存在 Node 中用 link list 串接起來(依照 starttime 再照 id 排序)

h) Outputstation_status: 依照 linklist 輸出 Insertstation_status

i) Insertuser_result: 儲存在 Node 中用 link list 串接起來(依照 userid 排序)

j) Outputuser_result: 依照 linklist 輸出 user_result

k) Inserttransfer_log: 儲存在 Node 中用 link list 串接起來(依照 userid 排序)

l) Outputtransfer_log: 依照 linklist 輸出 transfer_log

4.其他意見

a) 全勤可以加分嗎