

- 開發平台 : Code::Blocks
- 開發環境 : Windows10
- 使用語言 : C&C++

- 程式說明 :

- 功能 :

可以自檔案輸入各個 Process 之編號、CPU Burst、Arrival Time、及 Priority，並依這些資訊模擬 FCFS、RR、Preemptive SJF、Non-Preemptive SJF、Priority 等 CPU Scheduling。最後繪出 Gantt Chart，跟每個 Process 的 Turnaround Time 及 Waiting Time。

- Data structure :

資料存放處 為 struct 的一維陣列，因為做的是靜態陣列，size 為 100(已不可能達到的行數為主)，並配合 aP (存放資料總行數)，來確認是否下一行陣列還有資料。

而因第六題為全部排程處理，因此於每排程法做完時存放答案至 string 陣列，每一陣列存各個相對的 ID 答案。

Ex:

```
typedef struct Id
{
    // 存定義過的 ID
    int id;
    int cpuBurst;
    int arrivalTime;
    int priority;
    int waitingTime;    // 進入時間 - arrivalTime
    int turnaroundTime; // cpuBurst + waitingTime
    int used;           // 確認是否用過 CPU
    int uused;
} ID;

ID gProblem[100]; // 做事的陣列
ID gProblemTemp[100]; // 存放正確題目的陣列
```

```

int gTest;                // method
int gTime;                // time slice
int aP = 0;               // 文件總行數(有內容才算)
int gClock = 0;           // 目前時間

string gMap;              // 第 1~5 題的圖存放處
string gMapAnswer[100];   //第 6 題所有排程圖存放處
string gWaitingAnswer[100]; //第 6 題所有排程 Waiting Time
string gTurnaroundAnswer[100]; //第 6 題所有排程 Turnaround Time

```

➤ 流程：

程式流程 (由上到下)

```

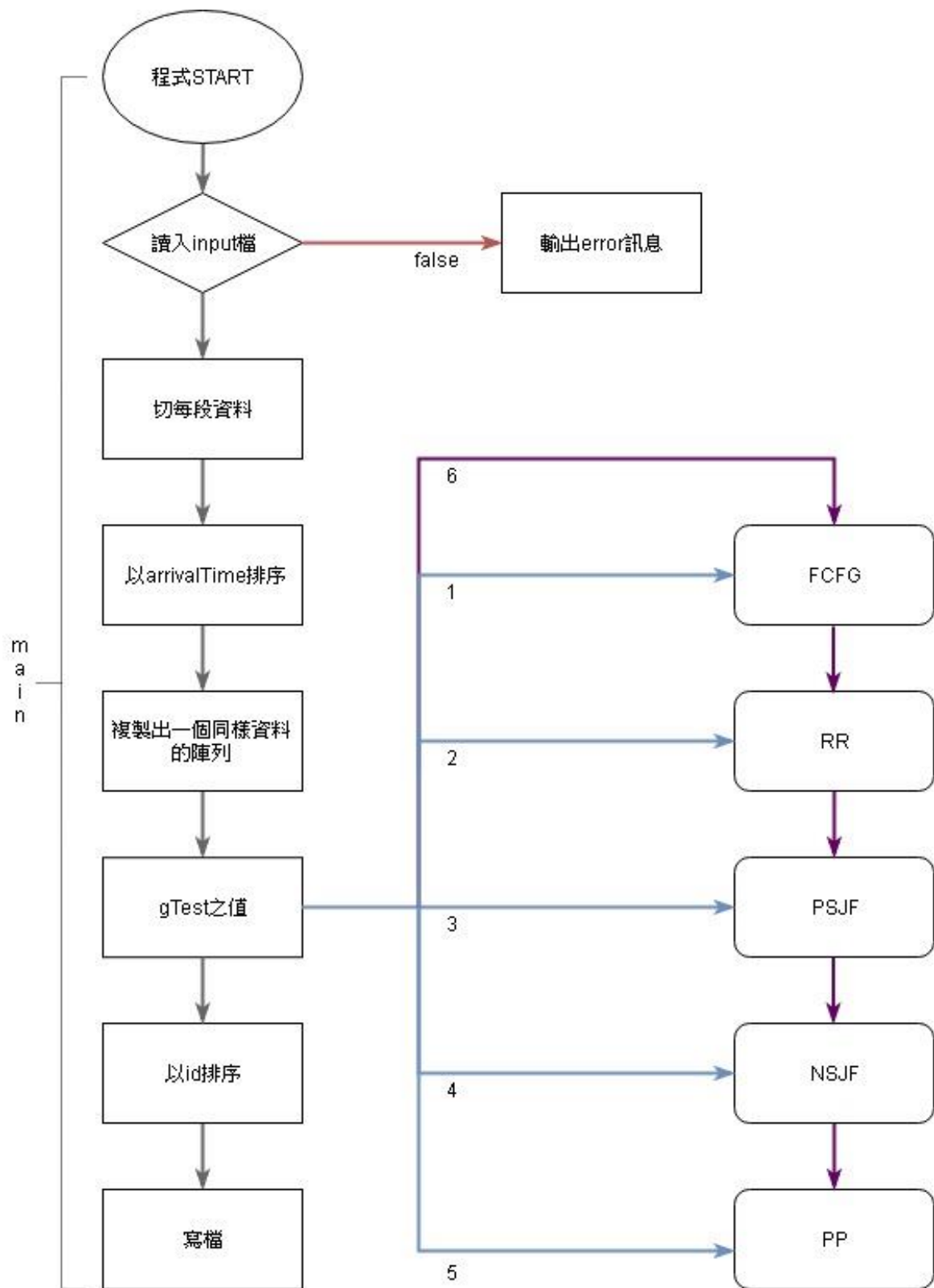
bool readProblem();        // 讀檔
void token();              // 切每段資料放入 struct
void arrivalTimeBubbleSort(); // arrivalTimeq 排序
void idBubbleSort();       // id 排序

//-----排程法-----//
FCFG();
RR();
PSJF();
NSJF();
PP();
//-----排程法-----//

void writeAnswer();        // 寫檔

```

程式流程圖



■ 未成功能：

1. 讀檔有限制 100 行，沒有做成 **vector** (可隨意增加行數)的模式。
2. 寫檔每個資料的間格是用 **tab**，這樣沒辦法讓所有數字對齊(因為答案的位元數不相同)，應該有其他方法使其對齊。

■ 遇到問題：

1. 寫檔的時候想很久，因為我所有排程都是使用同一個陣列，最後決定把答案全部先存到 **string** 裡，最後直接寫進去檔案中。
2. 在我程式碼的第 93 行，有一個 **string lala**，明明是完全沒用到的字串宣告，刪掉卻會影響到整個輸出資料(會變亂碼)，這部分的問題還未解決，也找不到原因。

■ 心得：

這次寫起來順手很多，只是最開始時想很久要 **python** 寫還是 **C++**，最後還是選擇了 **C++**，因為比較順手，寫法也比較能讓我直接聯想，雖然導致程式碼冗長，而且很多地方寫法很亂，但還好經過好幾天的努力，最終還是完成了這次的 **project** 測驗，而在所有排程法裡其實我卡最久的是在寫 **PSJF** 的時候，寫完後答案差超多，最後 **debug** 不了，我只好把每個過程的數據印出，一個個找問題所在之處，然後 **NSPF** 真的最簡單，我是直接拿 **RR** 去改，不到五分鐘就寫完了，這次藉由 **project** 讓我更加了解排程法的邏輯跟做法，算是學習的很大進步。