

銀行排隊問題

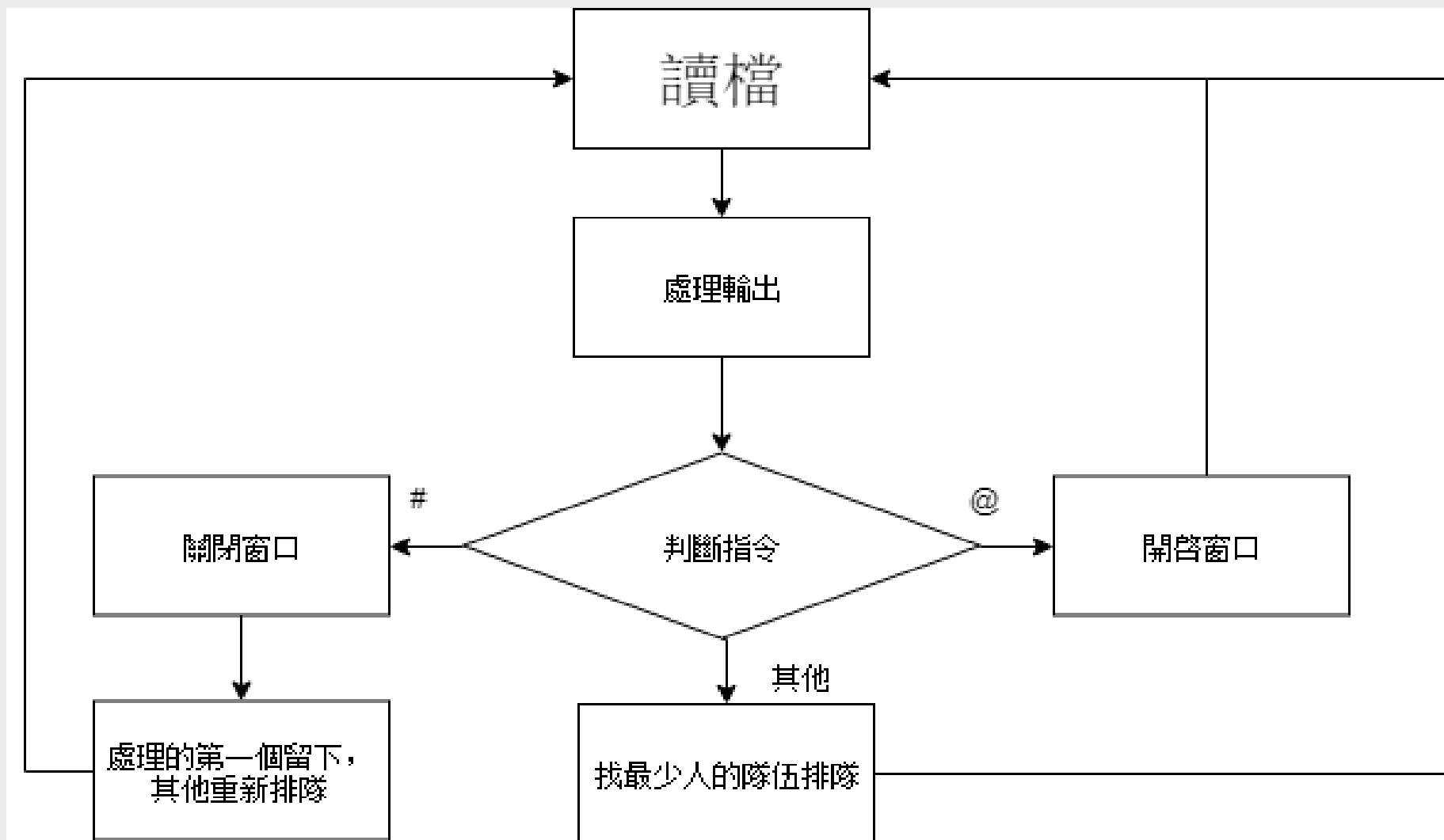
資訊二乙

結構決定

```
typedef struct ppl
{
    名字                                ( 名字自訂 )
    排隊的時間                          ( 名字自訂 )
    完成需要的時間                      ( 名字自訂 )
    離開的時間                          ( leaving time )
    排隊的窗口                          ( 名字自訂 )
    struct ppl *next                    ( 鏈節串列下一個節點 )
}ppl;

typedef struct counter
{
    ppl *head, *curr;                  ( 供鏈節串列使用 )
    窗口排隊人數                        ( ppl_num )
    開著還是關閉                        ( work )
    計算好第一個離開的時間            ( Processing time )
}counter;                             ( 名字自訂 )
```

流程圖



提醒重點

離開時間是由第一個人排隊的時間加上處理所需的時間。例如窗口排隊狀況為：

1 John 3 9

2 Alice 6 20

第一個人離開時間為自己排隊加上處理時間，第二個當然是直接把上一個離開時間加上自己處理時間。

John = $3+9 = 12$ (一開始的 processing time)

Alice = $12+20 = 32$ (更新的 processing time)

關窗口注意事項

排隊時窗口關閉，如果排去有人的窗口其實沒分別，因為用的都是處理時間加上上個人離開的時間。

可是如果排去剛剛開啓的窗口，因為沒有上個人，你的時間就會問題。

所以建議重新排隊的時間把自己的排隊時間改成用指令的關閉窗口的時間。

11 1

建議

寫的 function

- Void add(counter, 名字, 來的時間, 需要的時間)
功能：
 - (1) 找最少人的隊伍
 - (2) 建立鏈節串列 / 陣列
- Void remove()
把決定輸出的資料存起來，數字調整。

判斷功能

```
2
John 3 9
Mr.Brown 4 12
Alice 6 20
Jack 9 18
# 11 1
God 18 22
@ 22 1
GiGi 23 10
```

判斷藍色的部分：

- #：執行關窗的部分
- @：執行開窗的部分
- 其他：執行排隊的部分

```
2
John 3 9
Mr.Brown 4 12
Alice 6 20
Jack 9 18
# 11 1
God 18 22
@ 22 1
GiGi 23 10
```

判斷紅色的部分：

3 , 4 , 6 , 9 , 11 , 18 , 22 , 23

可以使用這些判斷到了哪個時間點處理輸出

(用void remove())

執行排隊

■ 用void add()找有開著而且最少人的隊伍

用來判斷的部分：

(1) 窗口排隊人數 (ppl_num)

(2) 開著還是關閉 (work)

work可以用 0 和 1 代表是否有在開啓狀態

記錄排隊人數比較方便找最少人的隊伍

■ 判斷是否為第一個排隊的

用來判斷的部分

ppl *head,*curr; (供鏈節串列使用)

if 是第一個人，Processing time=in + out，建立鏈節串列

else 新增鏈節串列，不用做其他事，等第一個人離開才計算下一個時間，因為中間可能重新排隊所以先計算沒意義。

執行開關窗

■ 開 (WORK=1)

直接打開下次執行排隊會是開啓狀態，允許排隊。

■ 關 (WORK=0)

把第一個人留下來，剩下的人都重新排隊。

重新排隊時間可以用接收指令的時間傳入void add(counter, 名字, 重排時間, 需要的時間)

處理輸出

- 每次讀一個指令時先判斷每個窗口的 Processing time 是否 \leq 紅色的數字，是的情況用 void remove 把人的離開資料存起來，有人離開時該隊伍要檢查多一次可能還是有人要離開。
- 把人的 leaving time = processing time，把資料存去準備輸出的陣列或鏈節串列，然後更新 processing time（就是 processing time + 排隊下一個人的需要時間）。

```
2  
John 3 9  
Mr.Brown 4 12  
Alice 6 20  
Jack 9 18  
# 11 1  
God 18 22  
@ 22 1  
GiGi 23 10
```

最後注意事項

- **讀檔** 到 **跳出迴圈** 還是有很多人還沒輸出，所以把所有窗口的人輸出，可以用void remove，無需條件把全部人輸出資料存起來。
- 然後輸出前照輸出時間排序一下，最後印在銀幕上。
- 此投影片只屬於建議與提醒，支持自由發揮。
- 請不要抄襲，被抄與抄襲者以0分計算。
- 無法完成請盡量寫，說明會和不會的部分還有遇到什麼問題，有交有分數加油！