

資料結構 TEST02 檢討

2014/11/20

Format

國立清華大學試卷

記		分	
1	2		
3	4		
5	6		
7	8		
9	10		
11	12		
13	14		
15	16		
17	18		
19	20		
總 分			

這是封面

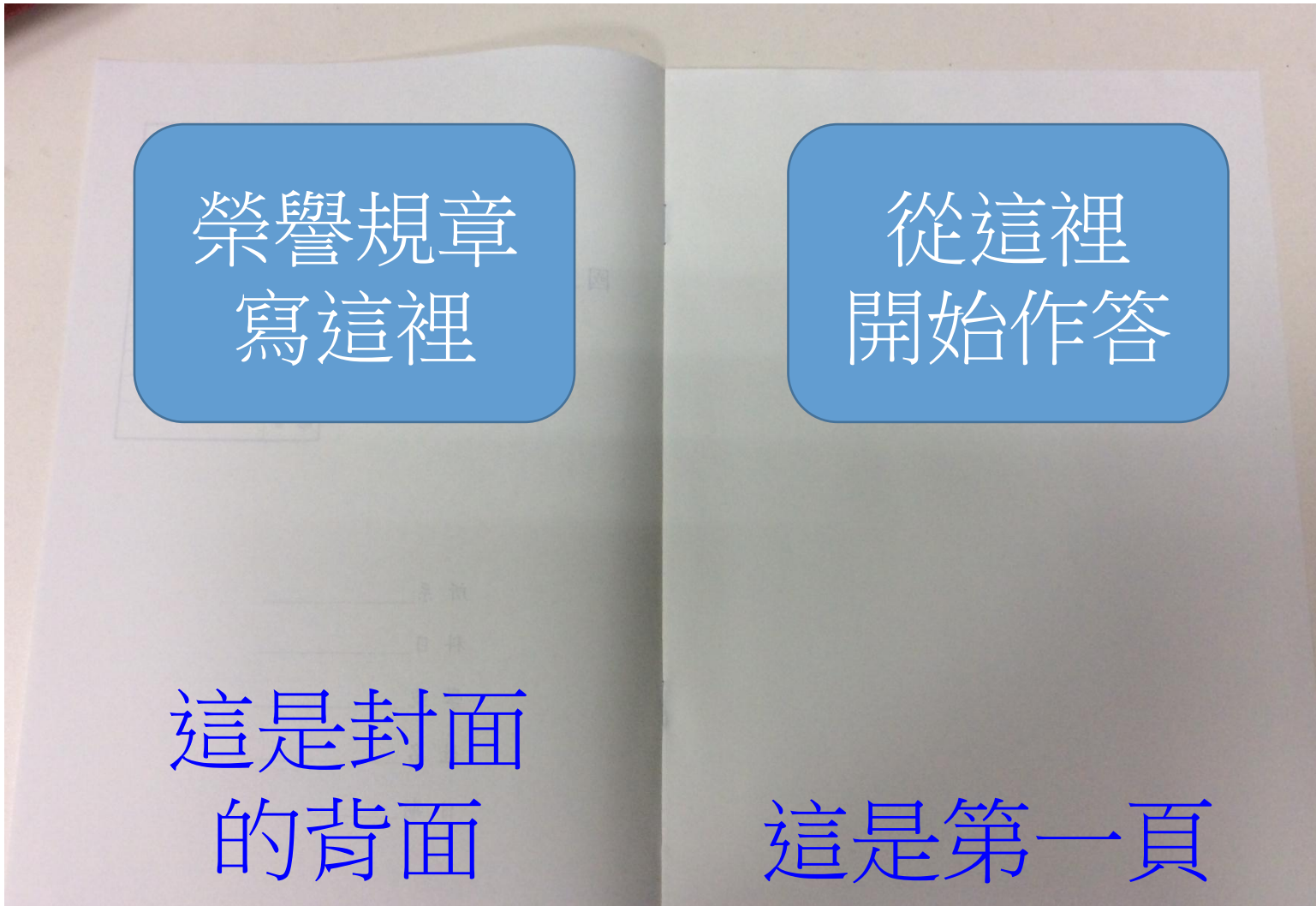
所 系_____

科 目_____

學 號_____

姓 名_____

日 期_____



榮譽規章
寫這裡

從這裡
開始作答

這是封面
的背面

這是第一頁

考試用榮譽規章

- 我將盡我所能完成此份試題，過程中絕不作弊，或以任何不公義之行為取得分數。

T1_circular.c

1.1

```
int qIsEmpty(struct Queue* Q) {  
    if((Q->rear+Q->qMax)%Q->qMax == (Q->front+Q->qMax)%Q->qMax) return 1;  
    else return 0;  
}  
//1.1 為什麼這麼函數不保持07_SandQ.c的寫法？  
//    如果維持之前的寫法if(Q->rear == Q->front) return 1  
//    會有什麼問題？(提示：一個環狀佇列至多只應該能存放qMax-1個物件)
```

- 本題送分
- 在這邊，使用 % 和直接用 Q->rear == Q->front 效果相同

Why 1.1

- 最多只能用 $qMax - 1$ 格的空間
- 如果用到滿的話...

index	-1	0	1	...	qMax-2	qMax-1
	↑ front					↑ rear

- 必須用 mod 才有辦法處理 -1 的情形 (才知道兩者相同)

Why 1.1 (Cont.)

- 另一個可能的問題
- 多用一個 int 存目前的 element 數?
 - 少浪費一格
 - 多用一個空間
 - 沒必要

1.2

```
if(Q->rear > Q->front) {  
    for(i = Q->rear; i > Q->front; i--) {  
        if(i == Q->rear) printf("rear ");  
        else printf(" ");  
        printf("%2d %d\n", Q->queue[i], i);  
//1.2 印出Q->queue[i]代表什麼？印出i代表什麼？  
// 他們能提供什麼資訊以方便顯示或debug？  
    }  
}
```

- i = 陣列索引值
- $Q \rightarrow \text{queue}[i]$ = 陣列第 i 格的內容
- 可以知道目前佇列內容是什麼

1.2 評分標準

- i 和 $Q \rightarrow \text{queue}[i]$ 的意義各 0.5 分
- i 寫成指標不給分
- i 寫成佇列位置或只有寫位置給分

1.3

```
if(Q->rear > Q->front) {  
    for(i = Q->rear; i > Q->front; i--) {  
        if(i == Q->rear) printf("rear ");  
        else printf(" ");  
        printf("%2d %d\n", Q->queue[i], i);  
//1.2 印出Q->queue[i]代表什麼？印出i代表什麼？  
// 他們能提供什麼資訊以方便顯示或debug？  
    }  
}  
else if(Q->rear == Q->front) {  
    printf("Queue is empty\n");  
}  
else {  
//1.3 這裡分了三段的流程控制，最後一段是else，  
// 也就是所有的支前沒有判斷的額外狀況。  
// 請問這個else的條件是什麼？
```

1.3 (Cont.)

- $\text{if} \rightarrow \text{rear} > \text{front}$
- $\text{else if} \rightarrow \text{rear} == \text{front}$
- $\text{else} \rightarrow \text{rear} < \text{front}$
- 也就是環狀佇列中 rear 超過尾巴繞回到前面的情況

1.3 評分標準

- 沒排除 $\text{rear} == \text{front}$ 扣 0.5 分
- 沒排除 $\text{rear} < \text{front}$ 扣 0.5 分

1.4

```
for(i = Q->rear; i != (Q->front+Q->qMax)%Q->qMax; i = (i+Q->qMax-1)%Q->qMax) {  
    if(i == Q->rear) printf("rear ");  
    else printf(" ");  
    printf("%2d %d\n", Q->queue[i], i);  
}
```

//1.4 這個迴圈在做什麼事情？

//1.5 這個迴圈能不能取代第60行的if區塊內的for迴圈？

// 用76~80行的迴圈取代61~67的迴圈，跑跑看。

//1.6 承上，如果可以取代而不改變結果，你覺得應不應該取代掉？

// 說明你的理由。

- 把佇列內的東西從 rear 到 front 依序印出來

1.4 評分標準

- 沒有提到依照順序扣 0.5 分
- 沒有提到印東西不給分
- 沒有結束條件扣 0.5 分
 - front+1 或 front 或結束...都可以

1.5

- 用這個迴圈換掉可不可以

```
for(i = Q->rear; i > Q->front; i--) {  
    if(i == Q->rear) printf("rear ");  
    else printf(" ");  
    printf("%2d %d\n", Q->queue[i], i);  
}
```

- 跑跑看，結果一樣就可以
- 結果一樣，所以可以

1.5 評分標準

- 本題是非題
 - 可以得 1 分
 - 不可以得 0 分

1.6

- 承上題，可以換掉的話該不該換掉
- 該，因為這樣就可以不用分 if 跟 else 兩種情形
- 不該，因為這樣跑起來比較慢
- 不該，因為這樣看起來比較複雜
- ...

1.6 評分標準

- 本題討論題
 - 理由合理足以說服助教都可以得 1 分
 - 無法說服助教或沒寫理由，但有寫答案得 0.5 分

1.7

```
int i;  
for(i = 0; i < 10; i++) {  
    add(&Q, i);  
}
```

```
//printQ(&Q);
```

//1.7 上面的迴圈試圖作10次的add，但是空間沒這麼大，
// 把上面的printQ註解拿掉然後紀錄結果。

Print Queue:

=====

rear	4	4
	3	3
	2	2
	1	1
	0	0

front

=====

1.8

```
for(i = 10; i < 15; i++){  
    printf("delete %d\n", del(&Q));  
    add(&Q, i);  
    //printQ(&Q);  
}
```

//1.8 把上面的printQ註解拿掉然後紀錄結果。

```
Print Queue:  
=====  
rear  10  5  
      4  4  
      3  3  
      2  2  
      1  1  
front  
=====
```

```
Print Queue:  
=====  
rear  11  0  
      10  5  
      4  4  
      3  3  
      2  2  
front  
=====
```

```
Print Queue:  
=====  
rear  12  1  
      11  0  
      10  5  
      4  4  
      3  3  
front  
=====
```

```
Print Queue:  
=====  
rear  13  2  
      12  1  
      11  0  
      10  5  
      4  4  
front  
=====
```

```
Print Queue:  
=====  
rear  14  3  
      13  2  
      12  1  
      11  0  
      10  5  
front  
=====
```

1.8 評分標準

- 每個部分 0.2 分，共 5 部分

1.7 & 1.8 注意事項

- 只要寫「打開註解」後「多印」的內容即可

1.9

```
void add(struct Queue* Q, int elem) {  
    if(qIsFull(Q)) return;  
    Q->rear = (Q->rear+1)%Q->qMax;  
    Q->queue[Q->rear] = elem;  
//1.9 我是否可以把上面兩行改成下面這兩行？  
//    Q->queue[(Q->rear+1)%Q->qMax] = elem;  
//    Q->rear = (Q->rear+1)%Q->qMax;  
//    把上面註解掉改成這兩行，然後跑跑看，是否和原來一樣？  
}
```

- 可以，和原本一樣
- 因為只要東西是放到 rear+1 那格，並且更新 rear 的值即可

1.10

```
int del(struct Queue* Q) {  
    if(qIsEmpty(Q)) return -1;  
    Q->front = (Q->front+1)%Q->qMax;  
    return Q->queue[Q->front];  
//1.10 我是否可以把上面兩行改成下面這兩行？  
//     return Q->queue[(Q->front+1)%Q->qMax];  
//     Q->front = (Q->front+1)%Q->qMax;  
//     把上面註解掉改成這兩行，然後跑跑看，是否和原來一樣？  
}
```

- 不可以，和原本不一樣
- 因為 return 之後的程式碼不會被執行，front 不會被更新到

1.9 & 1.10 評分標準

- 兩題都是是非題
 - 1.9 可以得 1 分、不可以得 0 分
 - 1.10 不可以得 1 分、可以得 0 分

T2_linkStack.c

2.1

```
printf("%d <- TOP\n", stack->sTop->num);  
struct StackNode* temp = stack->sTop->next;  
for(; temp != NULL; temp = temp->next) {  
//2.1 試著用你的方式解釋這個迴圈做的事情，  
// 必要時舉以圖例。  
    printf("%d\n", temp->num);  
}
```

- 從 top 的下一格開始，將 stack 內的元素依序印出來

2.1 評分標準

- 開始元素不是第二個元素扣 0.5 分
- 沒有寫印什麼(或寫錯)扣 0.5 分
- 其他不給分

2.2

```
void createS(struct Stack *stack, int sMax) {  
    stack->sMax = sMax;  
    stack->elemNum = 0;  
    stack->sTop = NULL;  
    //如果說把整數歸零是好習慣，那麼把指標歸NULL就是超級好習慣  
    //2.2 如果不要elemNum這個成員變數的話，  
    // 有什麼方法可以知道現在堆疊內的物件數？  
    // (提示：參考2.1)  
}
```

- 用數(尸 x ~ , count)的
- 最好可以用變數記下來而不是直接印出來

2.2 (Cont.)

```
int sSize(struct Stack *stack){  
    if(stack->sTop == NULL) return 0;  
    int element = 1; // 特別注意初始值為 1  
    struct StackNode* temp = stack->sTop->next;  
    for(; temp != NULL; temp = temp->next){  
        element++;  
    }  
    return element;  
}
```


2.2 評分標準

- 只要有表達用數(count)的意思就得 1 分
- 其他不給分
- 程式碼容易寫錯，盡量用文字描述 (這次沒扣分)

2.3

```
void push(struct Stack* stack, int elem) {  
    if(sIsFull(stack)) {  
        return;  
    }  
    stack->elemNum++;  
}
```

//2.3 上面這行的存在意義是什麼？

- 記錄目前堆疊內因為 push 的關係多了一個元素

2.3 評分標準

- 只寫 element 的意義但沒提到 push 會增加 1 扣 0.5 分
- 沒提到元素個數不給分

2.4

- 把 `stack->eleNum++` 註解掉

```
for(i = 0; i < 10; i++) {  
    push(&S, rand() % 10);  
}  
  
//printf(&S);
```

- 會不知道堆疊是否已經滿了
- 原本沒有註解掉之前，只有 4 個數字會被 `push` 進堆疊
- 註解掉之後，10 個數字會全部被 `push` 進堆疊

2.4 評分標準

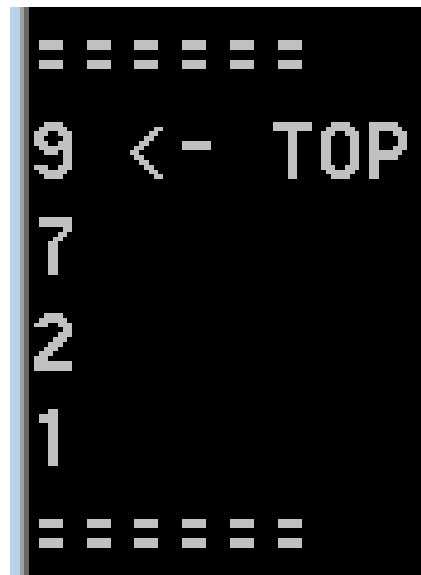
- 討論現象得 1 分
- 有討論原因更好
- 有解釋但沒討論 main 函數的程式碼扣 0.5 分

2.5

```
for(i = 0; i < 10; i++) {  
    push(&S, rand()%10);  
}
```

```
//printS(&S);
```

//2.5 把上面註解拿掉然後紀錄結果



```
=====
9 <- TOP
7
2
1
=====
```

2.5 評分標準

- 每次數字都不一樣，只要寫出 4 個 0~9 的數字都可以
- 只要寫「打開註解」後「多印」的內容即可

2.6

```
printf("pop %d\n", pop(&S));  
printf("pop %d\n", pop(&S));  
printf("pop %d\n", pop(&S));  
printf("pop %d\n", pop(&S));  
printf("pop %d\n", pop(&S));  
//printS(&S);
```

//2.6 把上面註解拿掉然後紀錄結果



```
=====  
<- TOP  
=====
```


2.6 評分標準

- 不要害怕他就是空的沒錯
- 只要寫「打開註解」後「多印」的內容即可
- 不用把 pop 那段也寫出來

休息 10 分鐘之後 TEST 3

- 使用 test2 的答案本接著寫
- 寫不夠可以到前面拿新的
- 用投影片放題目，請儘量往前坐
- 考試時間 50 分鐘，共 4 題
 - 每題放 10 分鐘，最後 10 分鐘自由瀏覽