

Homework 14

Objective :

熟悉使用 structure、pointer、queue 的用法

Exercise :(100%)

Queue 是一種常見的資料結構, 其資料的進出型態是先進先出。Queue 主要是由兩個函式操作, 一個是 enqueue, 另一個是 dequeue。請實作一個 Queue, 必須要包含二種功能, 二種功能分別寫成函式表示:

1. `void mydequeue (int num)`: 將傳入的數值 `num` 傳入到 Queue 中的正確位置, Queue 需要維持原來的排序。
2. `int myenqueue ()`: 將 Queue 中最先進入的元素拿掉並回傳裡面的數值, 該函式必須要做判斷, 當 Queue 中已經沒有元素可以拿出時, 必須要回傳錯誤訊息, 可以利用回傳 -1 來完成。

本次作業只需要從檔案 `input.txt` 讀入, 檔案一開始有一組數字, 必須先將數列傳入 Queue 中。接著檔案會有數次的 `enqueue` 或是 `dequeue`, 必須依照指令來做傳入數字或取出數字, 每做一個指令就印出當前 Queue 的狀態, 指令結束時直接結束程式即可。(程式請勿寫死, 測試實作作業時 `初始數字與個數`、`enqueue 次數`、`dequeue 次數`與 `push 輸入數字`會浮動)

Note: 檔名可以直接寫在程式中, 不用做使用者輸入檔名。


Note: enqueue 輸入數字只需判讀正負整數即可

Note: 判斷檔案結尾可以用 `feof()`, 成功時回傳 `true`。

Note: single 或 double link 皆可。

Note: 該次作業沒有要求要用 header 做。

Input.txt:

 input.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明

```
5
-5 3 9 12 7
Dequeue
Dequeue
Dequeue
Enqueue 15
Dequeue
Dequeue
Dequeue
Dequeue
Enqueue -10
Enqueue 9
Dequeue
Enqueue 123
Enqueue -15
Enqueue -108
Dequeue
Enqueue 12
Enqueue -6
```

Output:

```
Insert the original numbers.
Queue: -5 3 9 12 7
-----
Dequeue
Dequeue -5 from queue.
Queue: 3 9 12 7
-----
Dequeue
Dequeue 3 from queue.
Queue: 9 12 7
-----
Dequeue
Dequeue 9 from queue.
Queue: 12 7
-----
Enqueue 15
Enqueue 15 into queue.
Queue: 12 7 15
-----
Dequeue
Dequeue 12 from queue.
Queue: 7 15
-----
Dequeue
Dequeue 7 from queue.
Queue: 15
-----
Dequeue
Dequeue 15 from queue.
Queue:
-----
Dequeue
Nothing in queue
Queue:
-----
```

```
Enqueue -10
Enqueue -10 into queue.
Queue: -10
-----
Enqueue 9
Enqueue 9 into queue.
Queue: -10 9
-----
Dequeue
Dequeue -10 from queue.
Queue: 9
-----
Enqueue 123
Enqueue 123 into queue.
Queue: 9 123
-----
Enqueue -15
Enqueue -15 into queue.
Queue: 9 123 -15
-----
Enqueue -108
Enqueue -108 into queue.
Queue: 9 123 -15 -108
-----
Dequeue
Dequeue 9 from queue.
Queue: 123 -15 -108
-----
Enqueue 12
Enqueue 12 into queue.
Queue: 123 -15 -108 12
-----
Enqueue -6
Enqueue -6 into queue.
Queue: 123 -15 -108 12 -6
-----
Finish total input
```

Bonus:(50%):

請依照此次作業的內容與格式，實作不浪費記憶體空間的 Array 實作：

Circular Queue

通常以 Array 實作 Queue 時，需定義兩個變數來記住 Array 的 index：

1. 使用 front 記住 Queue 的開頭 index
2. 使用 back 記住 Queue 的尾端 index

進行以下操作：

Enqueue：若要新增資料，按照 Array 的 index 順序：0、1、2...，從 Queue 的 back 新增資料。

Dequeue：若要刪除資料，按照 Array 的 index 順序：0、1、2...，從 Queue 的 front 刪除資料。

但是 Dequeue 並沒有真正把 Array 的記憶體位置釋放，只是調整 front，使得 Queue「看起來」有刪除資料

Note: circular 的意思就是能夠繞回 Array 的前端，重複利用 Array 的記憶體空間

Note: Queue 最大儲存元素為 10 個

