Data Structure　　Homework2

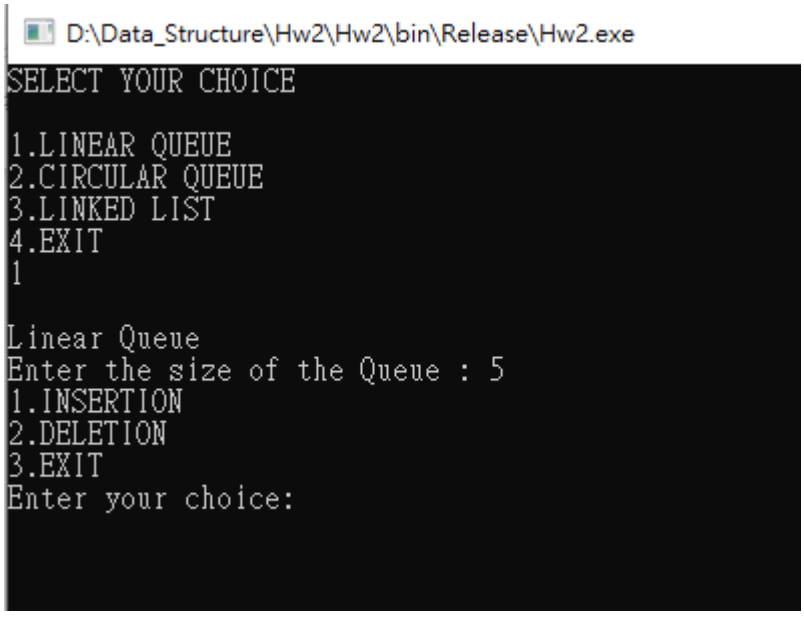
本次作業共分三小題，分別為用 array 實作出 Linked Queue 及 Circular Queue，以及用 linked list 實作 Circular Queue。

本次作業需要可以讓使用者自行輸入 Queue 的 size，且可 insert 或 delete 資料。並在每次 insert 或 delete 完資料後顯示 Queue 的內容及 Front、Rear。

若 Queue 為滿，需顯示 Queue is full；若 Queue 為空，則顯示 Queue is empty。

這次作業需要將 3 小題寫在同一份 code 裡，並且讓使用者自行選擇要選擇，並將其選擇列印出來。

程式範例 ：



※Example Input :
Enter the size : 5

Enter your choice : 1
Enter an element : 1

Enter your choice : 1
Enter an element : 2

Enter your choice : 1
Enter an element : 3

Enter your choice : 1
Enter an element : 4

Enter your choice : 1
Enter an element : 5

Enter your choice : 2

Enter your choice : 1
Enter an element : 5

※Example Output :

Linear Queue

Linear Queue : [] [1] [] [] []
Front = 0, Rear = 1

Linear Queue : [] [1] [2] [] []
Front = 0, Rear = 2

Linear Queue : [] [1] [2] [3] []
Front = 0, Rear = 3

Linear Queue : [] [1] [2] [3] [4]
Front = 0, Rear = 4

Queue is full!!
Linear Queue : [] [1] [2] [3] [4]
Front = 0, Rear = 4

Deleted element is : 1
Linear Queue : [] [] [2] [3] [4]
Front = 1, Rear = 4

Queue is full!!
Linear Queue : [] [] [2] [3] [4]
Front = 1, Rear = 4

Circular Queue 、 Linked List

Linear Queue : [] [1] [] [] []
Front = 0, Rear = 1

Circular Queue : [] [1] [2] [] []
Front = 0, Rear = 2

Circular Queue : [] [1] [2] [3] []
Front = 0, Rear = 3

Circular Queue : [] [1] [2] [3] [4]
Front = 0, Rear = 4

Queue is full!!
Circular Queue : [] [1] [2] [3] [4]
Front = 0, Rear = 4

Deleted element is : 1
Circular Queue : [] [] [2] [3] [4]
Front = 1, Rear = 4

Circular Queue : [5] [] [2] [3] [4]
Front = 1, Rear = 0

Notices

- ✓ 程式以C/C++語言為主
- ✓ 請附上說明檔，說明程式碼如何運作
- ✓ 請在程式碼中適時**加入註解**(未註解將酌量扣分)
- ✓ 作業須包含程式原始碼、執行檔(.exe)以及Word說明文件(ReadMe)，壓縮成.zip檔後上傳至moodle。
- ✓ 3題請寫在同一份code裡(未依照繳交將酌量扣分)
- ✓ 檔名HW2_學號_姓名.zip
- ✓ 請勿抄襲
- ✓ **Deadline: 4/19(日) 23:55 moodle繳交**