

程式設計補充作業 S13

Due: 2021/6/12 23:00

※注意事項：請依照課程網站內所公告之“作業檔案命名規則與規定”進行作業檔案命名以及繳交作業，未依照規定將斟酌扣分。

本次作業共有一題，為程式撰寫。

佇列(Queue)是一種有序的串列，插入以及刪除發生在不同端，也就是先進先出(First-In-First-Out)的串列。佇列只允許在後端（稱為 rear）進行插入操作，在前端（稱為 front）進行刪除操作。假設現在有一個佇列，包含了[10、4、16、20]，此時 delete_element()將回傳 20，之後佇列就變成 [10、4、16]；後來又 add_element(13)，佇列就變成[13、10、4、16]。如果沒有第一次的 adding，則無法 deleting 資料（因為佇列內部是空的）。

●作業題目：

請設計兩個類別，

1. 類別 Element 為存放佇列元素的類別，其包含兩個私有成員：整數(int data;)以及指向下一個元素的指標(Element* next;)，類別 Element 將以鏈結的方式來表示佇列。
2. 類別 Queue 為佇列的類別，其包含兩個私有成員，分別是指向佇列前端的指標(Element* front)以及指向佇列尾端的指標(Element* rear)。成員函式至少須包含加入新元素 add_element()以及刪除前端元素 delete_element()兩個函式、列印佇列所有資料的函式 print()、判斷佇列是否有元素的函式 is_empty()以及排序函式 sort()將佇列內的元素從小到大排序。

以上兩類別之私有成員請務必設為私有以保護其資料，若設為公有則不予計分。

※請實作類別 Queue 的除構子確保所有 Element 的空間將被正常釋放。

※請實作類別 Queue 的複製建構子，將佇列完整的複製。

※如果需要可於類別宣告其他的成員或成員函式。

提示：

佇列資料的儲存為鏈結的方式，在 adding 時請 new 一個新的 Element 至於佇列尾端，在 deleting 時則將前端的 Element 移出並將其值回傳主程式。

主程式請依以下順序輸出，delete_element ()、add_element (25)、delete_element ()、delete_element ()、add_element (30)、add_element (77) 、add_element (100) 、add_element (12)、sort()，每個步驟後都要列印佇列內容，排版請清晰明瞭，**若沒照格式輸出則不予計分。**

※請勿使用標準樣板函式庫(Standard Template Library)或與題目無關之巨集指令※