

一、規格要求，違反者以零分計！

- (a) 以 Dev-C++ 或 Code::Blocks 編譯與成功執行的 C/C++ 程式碼(.cpp/.c/.h/.hpp)，要有註解。
- (b) 任何一部分的程式碼都不得被偵測為抄襲。
- (c) 檔名限以「**DS1ex6_學號_學號**」開頭。

二、題目內容

整合下列任務在一個系統選單下，未整合、無法連續執行或沒有輸入防呆措施，都各扣 5 分。
若影響任務執行，該任務以零分計。

資料檔格式簡述：

- 取自網路的寶可夢資訊，第一列為欄位名稱，第二列起每列為一種寶可夢類型的資訊，共計 **13 個欄位**，以定位符號（'\t'）間隔，依「編號#」遞增排序。
- 資料檔名如 input601.txt，13 個欄位名稱和定義由左至右依序如下：(數值欄位皆為整數)

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
編號	名稱	類型 1	類型 2	總和	生命值	攻擊	防禦	特攻	特防	速度	世代	傳說與否

建樹必須遵守的原則：(每個任務違反一項各扣 5 分)

1. 預先不知道資料筆數，必須採用**動態陣列**或 **vector** 紀錄資料。
2. 樹狀結構禁止使用陣列，必須採用**動態指標**連結節點。
3. **鍵值相同的資料必須儲存於同一個節點上。**

(任務一) 建立二元搜尋樹

輸入：依「編號#」遞增排序的資料檔。

步驟：

- (1) 將資料全部存入動態陣列後，依「**編號#**」次序顯示每筆資料的 **6 個欄位**於螢幕上，包括：
編號、名稱、類型 1、生命值、攻擊、防禦。
 - (2) 依「編號#」次序讀取每筆資料，以「**生命值**」為鍵值建立二元搜尋樹，計算其樹高。
- 輸出：在螢幕上第一列顯示 **6 個欄位**名稱，第二列依「編號#」次序顯示每筆資料的 **6 個欄位**，左側加上一個流水序號以統計筆數，然後顯示「**生命值**」**二元搜尋樹的樹高**。

答問：上機時才將回覆寫入程式說明文件內。

其他必須遵守的原則：(每個任務違反一項各扣 5 分)

1. 搜尋資料禁止使用遍歷每個節點的暴力法，必須善用樹狀結構**避免走訪不必要的節點**。
2. 刪除搜尋樹的節點時，禁止只以標記加註，必須確實刪除該節點及調整樹狀結構。
3. 刪除節點有兩個子節點時，一律以**中序後繼者 (in-order successor)**進行置換。
4. 任務一建立樹狀結構之後，必須能夠持續**交替執行**任務二和任務三，不需要重新建樹。

(任務二) 單一欄位搜尋

// 單人組免做，完成別項任務方可獲得送分

輸入：任務一建立的二元搜尋樹，使用者指定的一個**最低門檻值**。

步驟：

- (1) 提示使用者指定一個**最低門檻值**，從「**生命值**」二元搜尋樹找出鍵值達到（**大於或等於**）最低門檻的所有資料（**可能是空集合**），並累計搜尋過程中**走訪過的節點數**。
 - (2) 依搜尋欄位值的遞減次序逐筆顯示搜尋結果於螢幕上，數值相等者再以「**編號#**」遞增排序，顯示每筆資料的**7 個欄位**包括：**編號、名稱、類型 1、總和、生命值、攻擊、防禦**。
 - (3) 建立樹狀結構之後，必須能夠反覆使用不同的**最低門檻值**執行，不用重新建樹。
- 輸出：將搜尋結果依「**生命值**」遞減次序逐筆顯示在螢幕上，第一列為**7 個欄位**名稱，第二列每筆資料各**7 個欄位**，左側加上一個流水序號以統計筆數，然後顯示**走訪節點數**。

答問：上機時才將回覆寫入程式說明文件內。

（任務三）刪除單一資料

輸入：任務一建立的二元搜尋樹。

步驟：

- (1) 從「**生命值**」二元搜尋樹找到**鍵值最大的一筆資料**，刪除該筆資料（**若是唯一資料，則刪除節點**），並維持原本的樹狀結構。
- (2) 於螢幕上顯示被刪除資料的所有欄位，然後移除該筆資料在**動態陣列中的對應內容**。
- (3) 每次刪除一筆資料後，於螢幕上再次顯示二元搜尋樹的樹高。刪除單筆資料及其節點之後，必須能夠反覆執行，逐步減少資料和樹高。

輸出：在螢幕上顯示被刪除資料的**13 個欄位**，第一列為欄位名稱，第二列是該筆資料的欄位值，然後顯示「**生命值**」二元搜尋樹的樹高。

挑戰：上機時微調任務的目標，現場修改原程式，並於截止期限內上傳程式碼。

三、參考範例，格式自訂，內容不可欠缺！

（任務一）建立二元搜尋樹

Input a file number [0: quit]: 601

	#	Name	Type 1	HP	Attack	Defense
[1]	1	Bulbasaur	Grass	45	49	49
[2]	2	Ivysaur	Grass	60	62	65
[3]	3	Venusaur	Grass	80	82	80
[4]	6	Charizard	Fire	78	84	78
[5]	7	Squirtle	Water	44	48	65
[6]	8	Wartortle	Water	59	63	80
[7]	9	Blastoise	Water	79	83	80
[8]	10	Caterpie	Bug	45	30	30
[9]	11	Metapod	Bug	50	20	50
[10]	13	Weedle	Bug	40	35	30
[11]	14	Kakuna	Bug	45	25	50
[12]	15	Beedrill	Bug	65	90	40

HP tree height = 5

(任務二) 單一欄位搜尋

Input a file number [0: quit]: 601

Threshold (a positive integer): 80

	#	Name	Type 1	Total	HP	Attack	Defense
[1]	3	Venusaur	Grass	525	80	82	80

Number of visited nodes = 3

Threshold (a positive integer): 65

	#	Name	Type 1	Total	HP	Attack	Defense
[1]	3	Venusaur	Grass	525	80	82	80
[2]	9	Blastoise	Water	530	79	83	80
[3]	6	Charizard	Fire	534	78	84	78
[4]	15	Beedrill	Bug	395	65	90	40

Number of visited nodes = 6

(任務三) 刪除單一資料

Input a file number [0: quit]: 601

// 1st deletion of the max

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
3	Venusaur	Grass	Poison	525	80	82	80	100	100	80	1	FALSE

HP tree height = 4

// 2nd deletion of the max

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
9	Blastoise	Water		530	79	83	80	85	105	78	1	FALSE

HP tree height = 4

// 3rd deletion of the max

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
6	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	FALSE

HP tree height = 4

