main:

讀入指令，若指令為0，程式執行結束；若指令為1，執行任務一；若指令為2，執行任務二；若指令為3，執行任務二；如果都不是，輸出"Command does not exit!"，請使用者再輸入一次指令。

任務一:

請使用者輸入檔案名稱，利用ReadFile()讀檔並把資料儲存至list中，每筆資料只擷取七個欄位；建立一個temp內容與list相同，呼叫SelectSort()依照學校代碼進行排序，完成後儲存至temp中，呼叫PrintFile()將資料依序顯示於螢幕上，以確保資料正確；分別呼叫SetBinaryTreeByGraduaters()、SetBinaryTreeBySchoolName()建立二元搜尋樹，起初樹是空的，直接將list中第0筆資料加入樹中，接下來資料處理方式是先利用一個迴圈找到該筆資料應該放置的位置，過程中判斷上學年度畢業人數或學校名稱是大於還是小於目前的節點，小於的話往左子樹繼續找，反之往右子樹搜尋，並紀錄樹高，找到位置後將list中的第i筆資料加入樹中，並分別設定該節點的leftChildPtr、rightChildPtr、parent，持續執行至list的N筆資料都處理完成，最後分別輸出二棵樹的樹高。

任務二:

分成兩個部分:搜尋畢業生數的二元樹、搜尋學校名稱的二元樹；前者請使用者輸入要搜尋的畢業生人數，用string讀入並利用atoi()轉換為整數，若是小於0，輸出”The input string is not a decimal number!”，反之呼叫SearchByGraduaters()進行搜尋，過程中是利用prorder的順序比較二元樹，若是該節點的畢業生人數大於或等於讀入的鍵值，就將該節點的內容放入temp中，最後呼叫PrintFile()將temp中的資料逐一顯示於螢幕上，如果temp是空的就輸出” There is no match!”；後者請使用者輸入要搜尋的學校名稱，利用SearchBySchool()進行搜尋，一樣使用prorder的順序比較，若是該節點的學校名稱與輸入的完全相同，就將該節點的內容放入temp中，最後呼叫PrintFile()將temp中的資料顯示於螢幕上，如果temp是空的就輸出” There is no match!”。

發現與心得

在讀檔時是利用計算tab數依序找出相對應的資料，並透過string的函式substr()將內容擷取下來加入至儲存資料的vector中，substr()的呼叫由於一開始讀進整行內容時是儲存在C字串中，要先透過函式assign()將C字串轉換成C++字串方可成功執行。

流程圖

提問

想要有效的降低樹高，想法是把二元樹的兩側平衡，方法是把每一次放進去的值設定為中間值，先將數據排序，以遞迴方式每次取資料的中間值放進二元樹，並將剩下資料分成兩段，再繼續找其餘資料的中間值依序放入。