#ifndef BSTlib\_h

#define BSTlib\_h

struct node

{

struct node \*left;

struct node \*right;

};

typedef struct node node\_t;

**以上為使用者可使用node\_t 的結構。**

**若使用此library，可使用到以下的功能**

（１）插入一個節點進BST（其插入的每個值必須符合BST的定義，即BST中的值不可有重複）

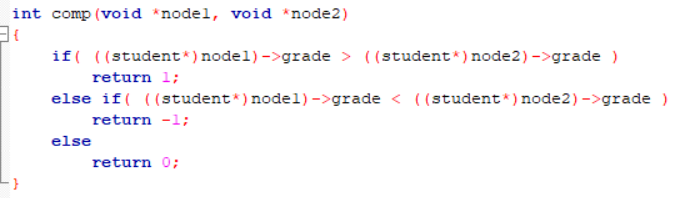
void insertNode(void \*element, node\_t \*\*root, int (\*cp)(void \*A, void \*B));

傳入值：  
第一個參數：型態為void，想要放入樹(root)節點的指標

第二個參數：型態為node\_t，樹根 (root)指標的位址

第三個參數：函式指標。

　　　　　　需使用者寫一個可以比較兩個node的函式並傳入，如：



　　　　　　（以上為欲比較學生成績的函式）  
 而需要其傳回的結果符合以下幾點

(1)當node1的值 > node2的值時，回傳1  
 (2)當node1的值 < node2的值時，回傳-1  
 (3)當node1的值 = node2的值時，回傳0

回傳值：無回傳值

（２）刪除BST中，與第一個傳入參數相等的值

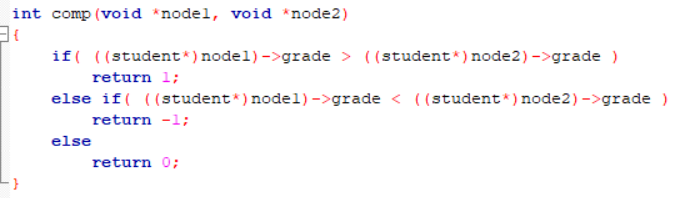
node\_t \*deleteNode(void \*target, node\_t \*\*root, int (\*cp)(void \*A, void \*B));

傳入值：  
第一個參數：型態為void，在tree(樹)中刪除與傳入參數相同的值的指標

第二個參數：型態為node\_t，樹根(root)指標的位址

第三個參數：函式指標。

　　　　　　需使用者寫一個可以比較兩個值(target&root中的值)的函式並傳入，如：



　　　　　　（以上為欲比較學生成績的函式）  
 而需要其傳回的結果符合以下幾點

(1)當node1的值 > node2的值時，回傳1  
 (2)當node1的值 < node2的值時，回傳-1  
 (3)當node1的值 = node2的值時，回傳0

回傳值：無回傳值

（３）找出BST中最小的鍵值

node\_t \*findMinNode(node\_t \*root);

傳入值：型態為node\_t，樹根(root)的指標

回傳值：型態為node\_t，指向最小值節點的指標

（４）找出BST中最大的鍵值

node\_t \*findMaxNode(node\_t \*root);

傳入值：型態為node\_t，樹根(root)的指標

回傳值：型態為node\_t，指向最小值節點的指標

（５）找出BST符合鍵值的節點

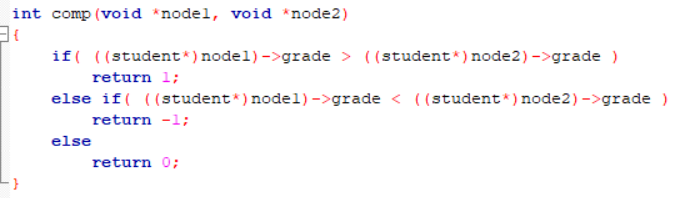
node\_t \*findNode(void \*target, node\_t \*root, int (\*cp)(void \*A, void \*B));

傳入值：  
第一個參數：型態為void，想找到在此樹中與此參數同值的節點的指標

第二個參數：型態為node\_t，樹根(root)指標的位址

第三個參數：函式指標。

　　　　　　需使用者寫一個可以比較兩個值(target&root中的值)的函式並傳入，如：



　　　　　　（以上為欲比較學生成績的函式）  
 而需要其傳回的結果符合以下幾點

(1)當node1的值 > node2的值時，回傳1  
 (2)當node1的值 < node2的值時，回傳-1  
 (3)當node1的值 = node2的值時，回傳0

回傳值：型態為node\_t，指向和target相同值，在root中的指標

（６）根據中序追蹤法，列印出BST中每個節點的內容

void inOrder(node\_t \*root, void(\*print\_node)(void\*));

傳入值：

第一個參數：型態為node\_t，傳入想要列印的樹的樹根(root)的指標

第二個參數：傳入一個指標函數，此函式可以將root中你想要的值列印出來

回傳值：無回傳值

（７）複製BST tree

void treeCopy(node\_t \*\*new\_tree, node\_t \*root, int struct\_node\_size);

傳入值：

第一個參數：傳入一個node\_t 型態，要複製到的樹的樹根的指標的位置

第二個參數：傳入一個node\_t型態，要被複製的樹的樹根的指標

第三個參數：傳入一個sizeof(root中節點的struct)後的int值

回傳值：無回傳值

（８）比較兩個BST是否相同

void treeEqual(node\_t \*tree1, node\_t \*tree2, int(\*cp)(void \*A, void \*B), int \*flag);

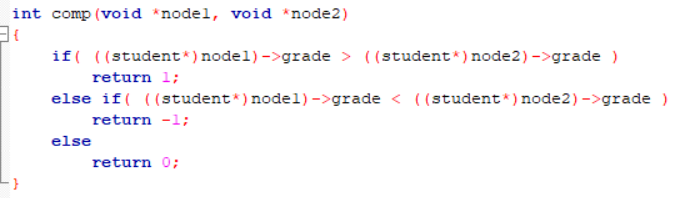
傳入值：

第一個參數：傳入一個node\_t 型態，要被比較的樹的樹根

第二個參數：傳入一個node\_t 型態，要被比較的樹的樹根

第三個參數：函式指標。

　　　　　　需使用者寫一個可以比較兩個值(tree1 &tree2中的值)的函式並傳入，如：



　　　　　　（以上為欲比較學生成績的函式）  
 而需要其傳回的結果符合以下幾點

(1)當node1的值 > node2的值時，回傳1  
 (2)當node1的值 < node2的值時，回傳-1  
 (3)當node1的值 = node2的值時，回傳0

第四個參數：需傳入一個int 型態，且其值為1的位址。

回傳值：無回傳值

（使者最後可以在自己的main中判斷flag的值，flag = 1表示兩個tree相等，flag = 0表示兩個tree不相等）  
#endif