

C 程式設計期末考

Date: 12/15

考試規定

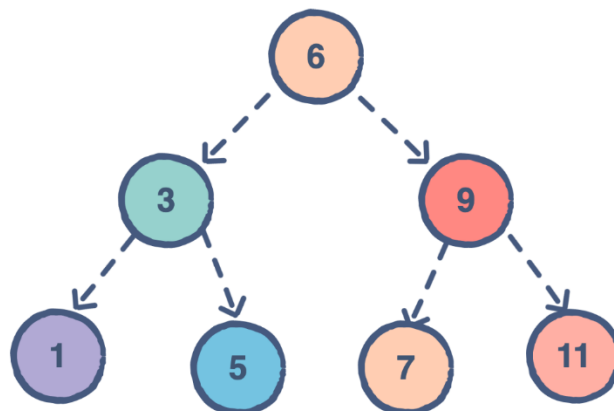
- 一律採用 Dev-C++ 進行考試，我們會統一以 Dev-C++ 作為批改標準。
- 批改會使用新安裝的 Dev-C++，若有無法編譯的情況，將視為是編譯錯誤。
- 考試答案請一律寫進同一檔案，並以學號.c 的方式命名。
- 考試中禁止查詢任何資料，僅允許使用網大上傳考試答案。
- 請注意繳交時間，過了繳交期限後不接受任何理由補交。
- 答案一律按照題目格式輸出，違者視情況斟酌扣分。
- 若有抄襲舞弊者，經證實後考試成績一律以 0 分計算。

考試題目

Binary search tree (BST) 為一種具有下列性質之 Binary tree：

若任意節點之左子樹不為空，則左子樹上所有節點的值，皆小於該節點的值；

若任意節點之右子樹不為空，則右子樹上所有節點的值，皆大於該節點的值。



An example of a binary search tree

請以 **Double Linked List** 實作一個 **BST**，並依序完成 a、b 小題：

- a. （ 10% ）請撰寫一個函式，先使用 `rand()` 亂數生成 15 個不重複的整數，

再以這 15 個整數建立一棵 **BST**。

※時間種子請設為 `srand(1)`

※亂數範圍：1 ~ 100

※Insert 操作僅可使用 `malloc()` 加入新的節點，請以函式完成

- b. （ 40% ）請提供一個功能選單，讓使用者能自由進行下列操作：

1. Insert：輸入一個整數後，於 **BST** 內新增一個節點
2. Delete：輸入一個整數後，於 **BST** 內刪除對應節點
3. Traversal：對 **BST** 分別進行 InOrder、PreOrder、PostOrder Traversal，
並將結果輸出至 console 上
4. Exit：切換至 Queue 選單

※Exit 以外的操作請以函式完成

※欲刪除的節點若同時有左子樹、右子樹，優先以左子樹之節點調整 **BST**

請實作兩個 **Stack**，以這兩個 **Stack** 實作一個 **Queue** 並完成 c 小題：

- c. （ 50% ）請提供一個功能選單，讓使用者能自由進行下列操作：

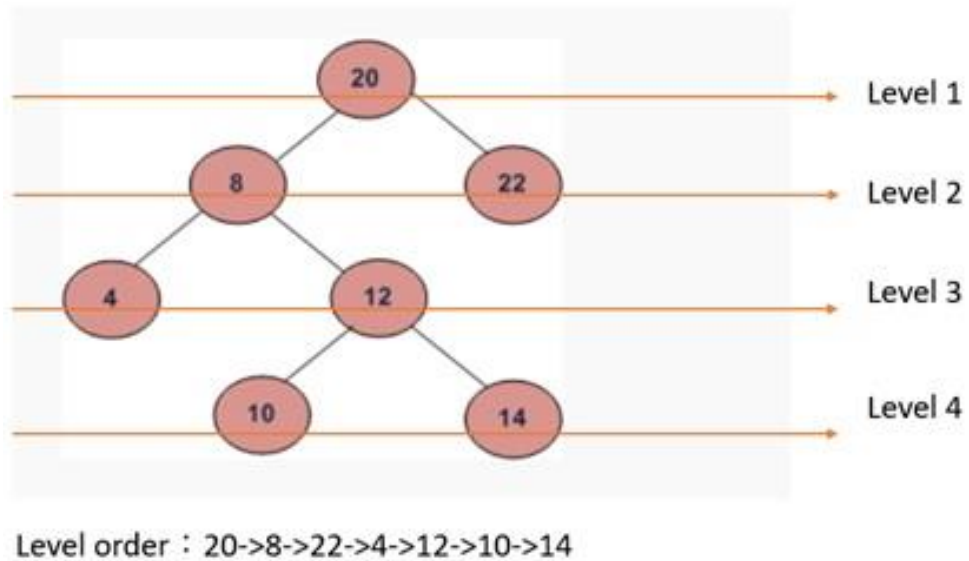
1. Enqueue：輸入一個整數後，於 **Queue** 內新增一個節點
2. Dequeue：於 **Queue** 內刪除一個節點
3. PrintQueue：將 **Queue** 內的資料輸出至 console 上

4. ClearQueue：將 Queue 內的資料全部清空

5. LevelOrder：以 Queue 對 b 小題的 BST 做 Level Order Traversal，

並將結果輸出至 console 上

Level Order Traversal：



6. Exit：切換至 BST 選單

※Exit 以外的操作請以函式完成

※Enqueue 操作僅可使用 `malloc()` 加入新的節點

※連續執行兩次 Exit 後即結束程式

輸出範例圖示：

```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
3
In-order traversal:
1->6->25->28->35->42->46->59->62->63->65->68->70->79->82->
Pre-order traversal:
42->35->1->25->6->28->68->59->46->63->62->65->70->79->82->
Post-order traversal:
6->28->25->1->35->46->62->65->63->59->82->79->70->68->42->
```

(圖一)將 a.小題的結果依三種 Traversal 方式輸出至 console

```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
1
Please enter the value to insert : 99
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
3
In-order traversal:
1->6->25->28->35->42->46->59->62->63->65->68->70->79->82->99->
Pre-order traversal:
42->35->1->25->6->28->68->59->46->63->62->65->70->79->82->99->
Post-order traversal:
6->28->25->1->35->46->62->65->63->59->99->82->79->70->68->42->
```

(圖二) Insert 99 至 tree，並且 Traversal 後輸出至 console

```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
2
Please enter the value to delete : 99
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
3
In-order traversal:
1->6->25->28->35->42->46->59->62->63->65->68->70->79->82->
Pre-order traversal:
42->35->1->25->6->28->68->59->46->63->62->65->70->79->82->
Post-order traversal:
6->28->25->1->35->46->62->65->63->59->82->79->70->68->42->
```

(圖三) Delete tree 中的 99，並且 Traversal 後輸出至 console

```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
4
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
4
Queue is empty now.
```

(圖四) 離開 tree mode，進入 Queue mode，並且清空 queue 內的資料

```

Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
4
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
1
Please enter the value to enqueue : 2
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
1
Please enter the value to enqueue : 4
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
3
*****
*       *
*       *
*   2   4   *
*       *
*       *
*       *
*****

```

(圖五) Enqueue 2 與 4，並將 queue 內之資料輸出至 console

```

Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
2
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
3
*****
*       *
*       *
*   4   *
*       *
*       *
*       *
*****

```

(圖六) Dequeue，並將 queue 內之資料輸出至 console

```

Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
4
Queue is empty now.
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
3
Queue is empty.

```

(圖七) 執行 LevelOrder 前，須將 Queue 內資料清空

```

Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
5
Level Order traversal of BST is :
42->35->68->1->59->70->25->46->63->79->6->28->62->65->82->

```

(圖八) 執行 LevelOrder，並將 LevelOrder 輸出至 console

```

Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
4
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
6
-----
Process exited after 5.605 seconds with return value 0
請按任意鍵繼續 . . . ■

```

(圖九) 如果連續執行兩次 Exit 將結束程式