# Final Exam

C COMPUTER PROGRAMMING ( I )

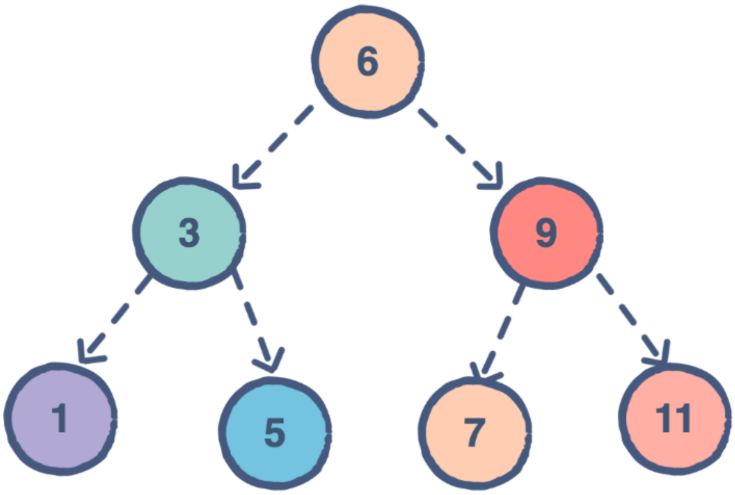
December 15, 2022

Exam rules

- Only Dev-C++ can be used for the exam.
- If your codes cannot be compiled by Dev-C++, it is considered as syntax error.
- Please write all your codes as a c source file named after your student ID.
- If your codes cannot output the desired result, your points will be deducted.
- Please check whether your codes can be compiled and output the desired result before submitting to National Sun Yat-sen Cyber University.
- No reason for late submission.
- It is forbidden to search for information during the exam.
- The cheaters will get zero point.

## Please design a program to complete the following questions:

A binary search tree (BST) is a binary tree with the following properties:
If the left subtree of any node is not empty, the values of all nodes on the left subtree are less than the value of the node.
If the right subtree of any node is not empty, the values of all nodes on the right subtree are greater than the value of the node.



An example of a binary search tree

**Please implement a BST with a double linked list, and complete question a & b:**

a. (10 points)

Please design a function that uses rand() to generate 15 non-repeating integers, and then built a BST with these 15 integers.

Note:

(1) Please set the seed of rand() using srand(1).

(2) Random number range: from 1 to 100.

(3) Please design a function to implement the insert operation.

(4) The insert operation can only add new nodes using malloc().

b. (40 points)

Please provide a function menu so that users can freely perform the following operations:

1. Insert : After entering an integer, add a new node in the BST.
2. Delete : After entering an integer, delete the corresponding node in the BST.
3. Traversal : Perform In-order, Preorder, Postorder traversal on the BST respectively, and output the result to console.
4. Exit : Switch to Queue menu.

Note:

(1) Operations other than Exit should be done with functions.

(2) If the deleted node has both left subtree and right subtree, the node of the left subtree is preferred to adjust the BST.

**Please implement two Stacks, use these two stacks to implement a queue and complete question c:**

c. (50 points)

Please provide a function menu so that users can freely perform the following operations:

1. Enqueue : After entering an integer, add a new node in the queue.
2. Dequeue : Delete the first node in the queue.
3. PrintQueue : Output the data in the queue to the console.
4. ClearQueue : Clear all the data in the queue.
5. LevelOrder : Perform Level order traversal on the BST of question b, and output the result to console.
6. Exit : Switch to BST menu.

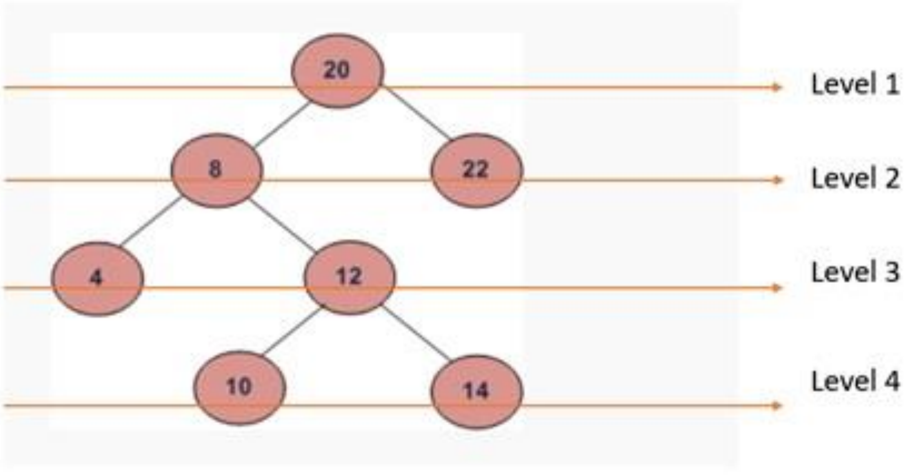
Note:

(1) Operations other than Exit should be done with functions.

(2) The enqueue operation can only add new nodes using malloc().

(3) You have to clear whole queue before perfom the levelorder function.

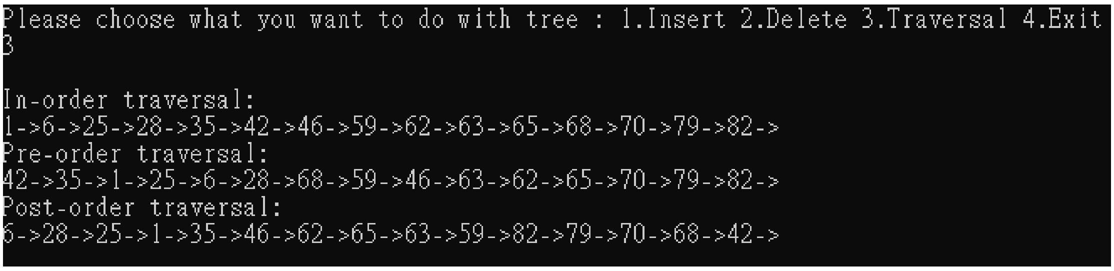
(4) The result of level order traversal is as follows:



Level order：20->8->22->4->12->10->14

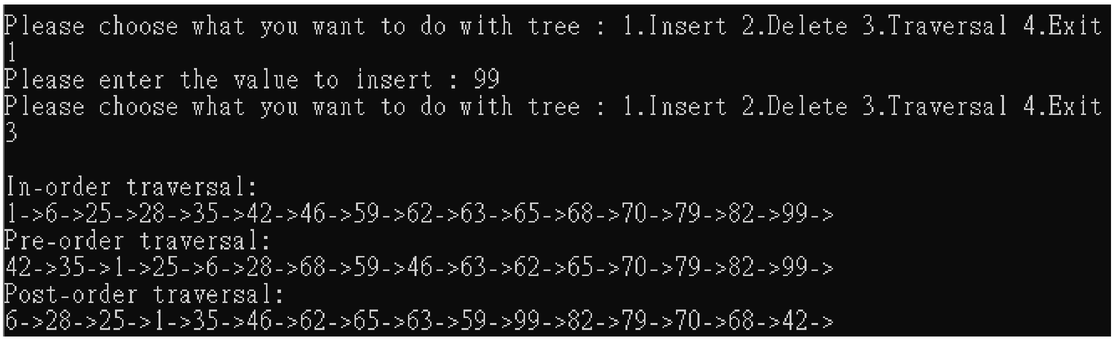
(5) Executing Exit function twice in a row will end your program.

**Sample Output**



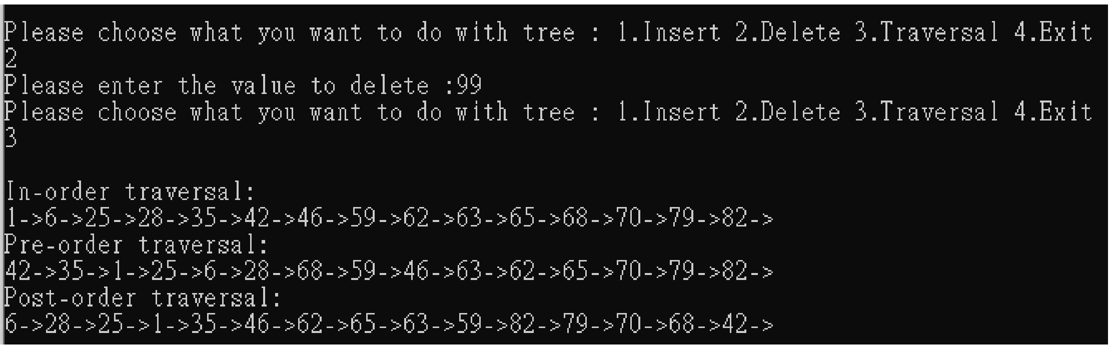
```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
3

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->
```

Figure.1 Output three traversal results to console



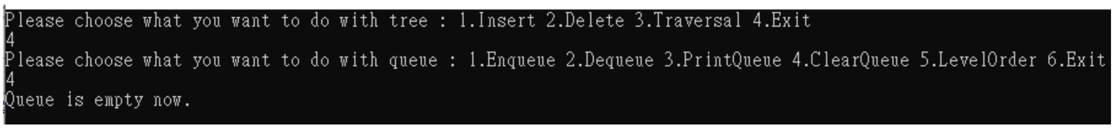
```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
1
Please enter the value to insert : 99
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
3

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->
```

Figure.2 Output three traversal results to console after inserted 99 to BST.



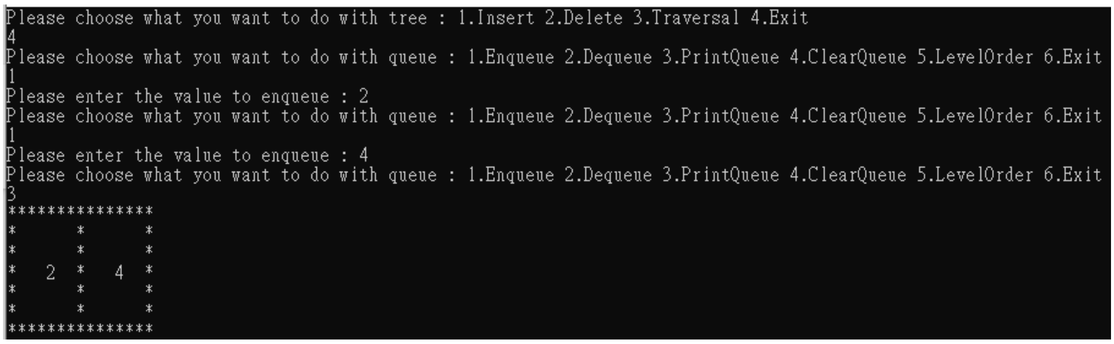
```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
2
Please enter the value to delete :99
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
3

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->
```

Figure.3 Output three traversal results to console after deleted 99 to BST.



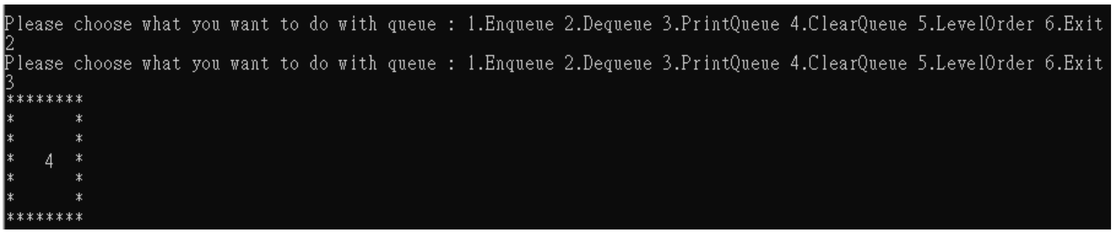
```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
4
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
4
Queue is empty now.
```

Figure.4 Switch to Queue menu and clear all the data in the queue.



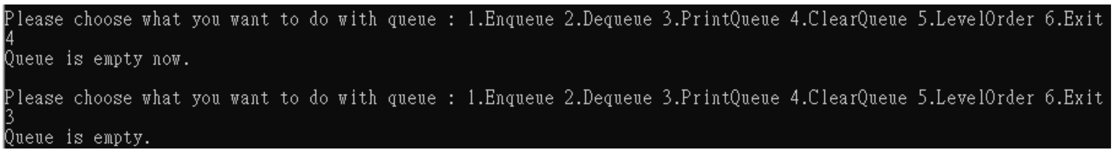
```
Please choose what you want to do with tree : 1.Insert 2.Delete 3.Traversal 4.Exit
4
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
1
Please enter the value to enqueue : 2
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue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

Figure.5 Enqueue 2 and 4, then output the data in the queue to the console.



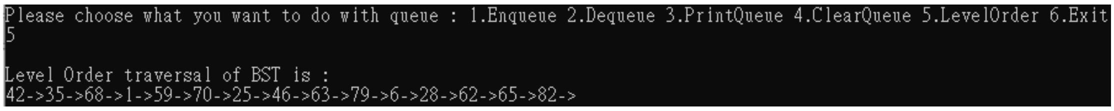
```
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
2
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
3
********
*      *
*      *
*   4  *
*      *
*      *
********
```

Figure.6 Dequeue, then output the data in the queue to the console.



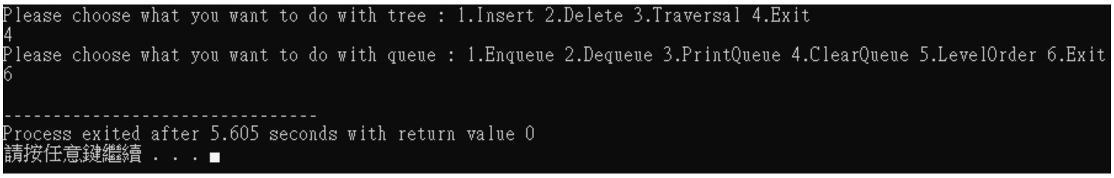
```
Please choose what you want to do with queue : 1.Enqueue 2.Dequeue 3.PrintQueue 4.ClearQueue 5.LevelOrder 6.Exit
4
Queue is empty now.

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.
```

Figure.7 Clear all the data in the queue before performing LevelOrder function.



```
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->
```

Figure.8 Perform Level order traversal on the BST and output the result to 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 . . . ■
```

Figure.9 Executing Exit function twice in a row will end your program.