***Quiz 3  
Instructor: Albert Hambardzumyan***

***Duration: 1h 30min***

***ADT***

***1 (25 points).*** Implement the concept of Stack in the given below ways by including the following methods:

**public class** Stack {

**private Node** head;   
 …

}  
  
The constructor with no arguments;

**boolean** isEmpty(); // check whether the stack is empty or not  
 **void** pop(); // deletes the top node  
 **void** push(int value); // add a new node with specified value   
 **int top**(); // returns the value of the top node  
  
Implement Node class in your own preference.

Write short test for each of your implementation.

***2 (25 points).*** Implement the concept of Queue in the given below ways by including the following methods:

**public class** Queue {

**private Node** head, tail;  
 …

}  
  
The constructor with no arguments;

**boolean** isEmpty(); // check whether the stack is empty or not  
 **void** pop(); // deletes the top node  
 **void** push(int value); // add a new node with specified value   
 **int top**();// returns the value of the top node  
  
Implement Node class in your own preference.

Write short test for each of your implementation.

***3 (25 points).*** Implement the concept of LinkedList in the given below ways by including the following methods:

**public class** LinkedList {

**private Node** head;

**private int** length;  
 …

}  
  
The constructor with no arguments;

**boolean** isEmpty(); // check whether the stack is empty or not  
 **int** get\_Length(); // return the length of the linked list  
 **void** remove(int from); // deletes the node from the specified position  
 **void** insert (int at, int value); // add a new node with specified value in specified position  
 **int retrieve** (int from); // returns the value of the node for the specified position   
  
Implement Node class in your own preference.

Write short test for each of your implementation.

***4 (25 points).*** Implement the concept of BST in the given below ways by including the following methods:

**public class** BST {

**private Node** root;  
 …

}  
  
The constructor with no arguments;

**int** min(); // return the min value in the tree  
 **int** max(); // return the max value in the tree  
 **Node** search(int value); // returns the node having the specified value  
 **void** inorder(); // inorder traversal of the tree  
 **void** preorder(); // preorder traversal of the tree  
 **void** postorder(); // postorder traversal of the tree  
 **void** insert(int value); // inserts the specified value in proper position  
 **void** delete(int value); // deletes the specified value in proper position  
 **void** readFromArray(int[] array); // overrides the tree by sorted array values  
  
Implement Node class in your own preference.   
Note, for above methods you may need private methods, such as wrappers.

Write short test for each of your implementation.