Binary Search Tree will be covered in the exam.

I. What’s the output of the following program? [20 pts]

```java
... 
public static void main(String[] args) 
{ 
    print(4); 
} 

void print(int num) 
{ 
    if(num > 1) 
    { 
        print(num-2); 
        System.out.println(num); 
        print(num-1); 
    } 
    else 
    { 
        System.out.print(num); 
    } 
} 
```

II. Give a solution to evaluate Fib without recursion. [20 pts]

```java
int Fib(int num) 
{ 
    if(num <= 1) 
    return num; 
    else return (Fib(num-2) + Fib(num-1)); 
} 
```
III. Give the recursive implementation of delete for sorted linked list. [30 pts]

class Node
{
    int value;  // holds the int value of the node
    Node next;  // holds the successor
}

class LinkList
{
    Node head;

    // if it is deleted return true
    // return false otherwise
    public boolean delete(int item)
    {
    }
}

IV. Give the recursive implementation of insert for BST. [20 pts]

class BinarySearchTree<T extends Comparable>
V. Given a post-fix expression, convert it to an in-fix expression. You may add as many parenthesis as you may need. (You may assume Stack is available)[25 pts]

    // convert the post-fix expression to an in-fix expr
    // for example: 5 6 7 + * will be converted to
    // (5 * (6 + 7))
    String convert(String postExpr)
    {
    }
VI. Give the implementation of LinkList based Queue. [25 pts]

class Queue<T>
{
    class Node
    {
        T value; // all values are positive
        Node next;
    }

    Node head;
    Node tail;
    // create an empty stack
    public Queue()
    {

    }

    // enqueue item
    public enqueue(T item)
    {

// if the queue is empty return null
public T dequeue()
{

}
}

VII. What are the advantage of using LinkList implementation of Stack over array-based implementation? [20 pts]

VIII. Give the recursive implementation of the following code (you should not use while, do while or for loops in your implementation). [20 pts]

```java
int SquareFact(int num)
{
    int result = 1;
    for(int i = 1; i <= num; i++)
    {
        result = result * i * i;
    }
    return result;
}
```
IX. What’s the result of the following operations for an empty binary search tree at the beginning. [20 pts]
I 10, I 8, I 25, I 9, I 6, I 7, I 22, I 33, I 11, D 9, D 10