1. a)Write a recursive method print the value from n to 1. (n is a positive integer). Example: n=5 should print out 10 8 6 4 2 1 [3 mins]

Public void print2Nums(int n){

If(n<=0){

Print(“1”);

}

Print(n\*2);

Print2Nums(n-1);

}

b) Write a recursive method that takes two input x and y, and returns the value that x is raise to power of y. (x and y are non-negative) Example: pow(2,3) returns 8 [3 mins]

2. Write a recursive method that count the occurrence of “1” in a string. Example: “213243453213” will return 2. [5 mins]

Public int countOnes(String s){

If(s.equals(“”)){

return 0;

}

char firstChar = s.charAt(0);

leftover String = s.substring(1);

if(firstChar == ‘1’){

Return 1 + countOnes(leftover);

}else{

Return countOnes(leftover);

}

}

3. Given two non-negative integers, write a recursive method that returns their greatest common divisor(GCD) and use it to find their least common multiple (LCM). [5 mins]

Public int GCD(int x, int y){

If(x>=y){

If(x % y ==0){

Return y;

}

Else{

Return GCF(x, x % y)

}

}else if(y %x == 0){

Else{

Return x;

}

Return GCF(x, y % x);

}

}

Public LCM(){}

4. Write a program that reverse a string using recursion.[7 mins]

Public String rev(String s){

If(String.equals(“”)){

Return “”;

}

Char current = s.charAt(0);

String leftover= s.substring(1);

Return rev(leftover) + current;

}

5. Given 2 linked lists L1 and L2, return a linked list L which contains all the elements of L1 and L2 in alternating order. You must do this recursively. [7 mins]

Example:

L1 = 1->2->3->4->5

L2 = 11->12->13->14->15

L = 1->11->2->12->3->13->4->14->5->15

Public ListNode alternate(ListNode L1, ListNode L2){

If(L1 == null && L2 == null){

Return null;

}else if(L1 == null){

Return L2;

}else if(L2 == null){

Return L1;

}

ListNode L1Next = L1.next();

ListNode L2Next = L2.next();

L1.next = L2;

L2.next = alternate(L1Next, L2Next);

Return L1;

}

6. a) Write a recursive method that returns the sum of natural numbers up to n. The recursive function should only accept a single integer(n) as a parameter. Is it possible to make the function tail recursive? [5 min]

Public int sum(int n){

If(n == 0){

Return 0;

}else{

Return n + sum(n-1);

}

}

b) Supposing you could pass more than one parameter, write a recursive function which returns the sum of the first ‘n’ natural numbers such that it is tail recursive. [5 min]

Sum(int n, int i){

If(i==n){

Return n;

}else{

Return (I + sum(n, i+1);

}

}

7. Implement Binary Search on an array with positive integers. Return its index in this array. Return -1 if the target is not found. [10 mins]

Public int binSearch(Array a, int target){

If(a.length == 0){

Return -1;

}elseif(a.length == 1){

If(a[0] == target)

}

}