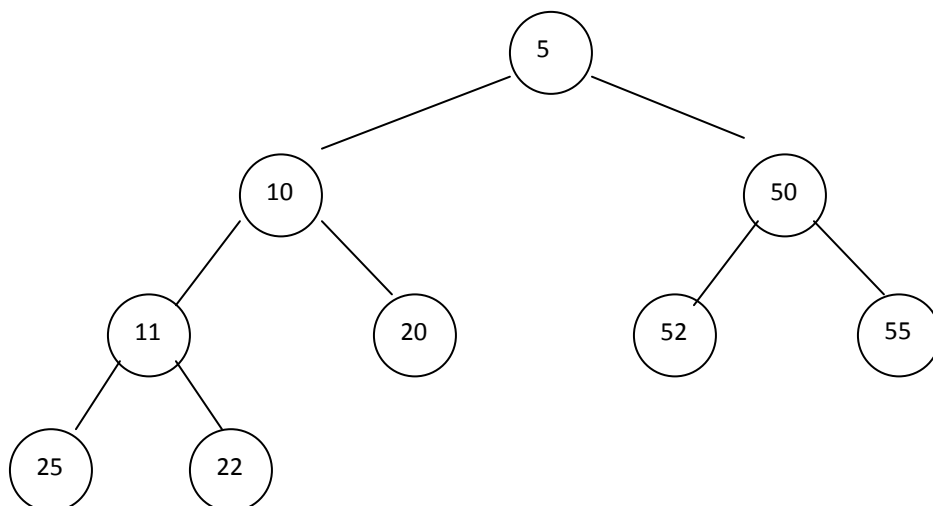


**Çankaya University Department of Computer Engineering**

**CENG 218 Data Structures Final (2008-2009 Spring 08.06.2009) Time allowed: 120 minutes**

1. A stack can be used to recognize a certain types of patterns. Consider the pattern **STRING1#STRING2** where neither string contains # and STRING2 must be the reverse of STRING1. For instance, the string **123&^a#a^&321** matches the patterns but the string **a2qd#dq3a** does not. Write a program that reads string and indicates whether each string matches the pattern. (15p)
2. Assume L is a list of integers(Singly Linked List). Write a function `Void OddEven(const LinkedList<int>&L, LinkedList <int>&L1 , LinkedList <int>&L2);` That takes a linked list L and creates two new lists L1 and L2. L1 contains the nodes of L whose data values are an odd number. L2 contains the nodes whose data values are even numbers. (15p)
3. Write a recursive `deleteRear` method which will remove the last element of linked list. (assume Linked list class maintains only head pointer). Method should return true if last node successfully delted false otherwise(10p)
4. Write `Copy Tree` (copy constructor for binary tree ) method for binary tree structure. (10p)  
**Template<Class T>**  
**Void BinaryTree<T>:: BinaryTree (const BinaryTree<T> & t)**
5. Write the recursive `PrintLeafs` function that prints only the data part of leaf of a binary tree .(10p)  
**Template<Class T>**  
**Void BinaryTree<T>::printLeafs(Tnode<T> \*r)**
6. Write a method called `sum_nodes` that returns the sum of the values in linked list.  
a)Write an iterative (non-recursive) version of `sum_nodes`. (5P)  
b)Write a recursive version of `sum_nodes`. (15P)
7. Assume we are dealing with the following array: **[7, 5, 2, 13, 14, 12, 20]** What will the array look like after 2 iteration of the outer loop for each of bubble sort, insertion sort, heap sort methods?(ascending order) **(15p)**
8. Heap(minimum). Consider following heap structure. Assume heap is stored in array having size 12.  
a)Show array content that corresponds to following heap structure.(5p)  
b)insert 8 and draw the final heap. (5p)  
c) apply one delete operation and draw the final heap(5p)



```

template<class T>
class Node {
public :
    T info;
    IntNode *next;
    IntNode(T el, IntNode *ptr = NULL){
        info = el;
        next = ptr;
    }
};

template<class T>
class SList {
public :
    SList(){
        head = tail = NULL;
    }
    ....
    ...
    ....
    ....
private :
    Node<T> *head;
};

```

```

template<class T>
class TNode {
public :
    T info;
    TNode *lchild,*rchild;
    TNode (T el, TNode *lc = 0, TNode *rc = 0){
        info = el;
        lchild=lc;
        rchild =rc;
    }
};

template<class T>
class BTree {
public :
    SList(){
        Root=NULL0;
    }
    ....
    ...
    ....
    ....
private :
    TNode<T> *Root;
};

```