
# Mohammad Ali Jinnah University

Chartered by Government of Sindh - Recognized by HEC

## Lab Project Proposal

**Project title:** Phone Contacts Management System

**Subject:** Data Structures and Algorithms Lab (CS 2511)

**Section:** AM

**Teacher:** MUHAMMAD MUBASHIR KHAN

**Date:** Tuesday, June 8, 2020

## Team Member Details:

1. Anas Ali Khan (SP20-BSCS-0015).
2. Hasan Ahmed (SP20-BSCS-0028).
3. Taimoor Manzoor (SP20-BSCS-0021).

**Project title:** Phone Contacts Management System

## Details:

It is a Smart Contact Management System, which lets you manipulate data of your Contact Library as you want with various functionalities. I have tried to put numerous functions into one single Application for greater productivity.

**KEY FEATURES**

1. Contact Addition(using LinkList)
2. Contact Deletion (using Stack, Queue, index,LinkList)
3. Contact Finder(using Tree)
4. Viewable Contact Library(using LinkList ,Tree)

**TECHNOLOGY USED**

1. Java (We have used the core concepts of Java along with the Data Structure-Algorithms, like; Linked list, Stack, Queue, Trees).

**CODE AND OUTPUT SCREENSHOT:**

- **INSERT FUNCTION:**

```
public listed insert(listed list, String a,String b)
{
    //it is intailzed the new Node
    Node new_Node = new Node(a,b);
    //head==null
    if(list.head==null)
    {
        list.head = new_Node;
    }
    //otherwise it will run
    else
    {
        //last value will assign to head
        last = list.head;
        //it traversing the whole linked list and insert in last
        while(last.next!= null)
        {
        last=last.next;
        }
        last.next= new_Node;

    }
    return list;
}
```

```
Enter Contact Name Phone number
Mike
Enter a Cell Phone number
030021222233
-------------------------
03002661270 Hasan
03002661270 Anas
03002661270 Taimoor
030021222233 Mike
```

- **Delete AT INDEX FUNCTION:**

```
public listed Deletion_at_index(listed list,int b)
{
        pre=head;
        int count =1;
        //if index == count (1==1) then it will run
        if(b==count)
        {
           Node h = head.next;
           head=null;
           head=h;
        }
        else
        {
        //finding the index value
        while(count < b-1)
        {
            pre = pre.next;
            count++;
        }
        //here we are doing swapping
        curr = pre.next;
        pre.next=curr.next;
        curr=null;}
        return list;
    }
```

```
03002661270 Hasan
03002661270 Anas
03002661270 Taimoor
We Have three Option for the deletion of Contact Number
Deletion At Index | Press 1
Delete At Stack Format  | Press 2
Delete At Queue Format | Press 3
1
Enter the index of contact number
2
After Update
03002661270 Hasan
03002661270 Taimoor
```

- **Delete AT End FUNCTION:**

```
public listed DeleteEnd(listed list)
{

        Scanner sc = new Scanner(System.in);
        //HEAD==NULL THEN IT WILL RUN
        if(list.head==null)
        {
            System.out.println("Empty");
          }
        else
        {
            // Find the second last node
            Node second_last = list.head;
            while (second_last.next.next != null)
            {
                second_last = second_last.next;
            }
            // Change next of second last
            second_last.next = null;
        }
       return list;
}
```

```
03002661270 Hasan
03002661270 Anas
03002661270 Taimoor
We Have three Option for the deletion of Contact Number
Deletion At Index | Press 1
Delete At Stack Format  | Press 2
Delete At Queue Format | Press 3
2
After Update
03002661270 Hasan
03002661270 Anas
```

- **Delete AT First FUNCTION:**

```
public void DeleteFirst()
{
    //if head==null then it will did not delete any element
    if (head == null)
    {
    return;
    }
    // here it delete the element
    else
    {
          if (head == last)
          {
                head = null;
                last = null;
          }
          else
          {
                  Node h = head.next;
                  head=null;
                  head=h;
          }
    }
}
```

```
03002661270 Hasan
03002661270 Anas
03002661270 Taimoor
We Have three Option for the deletion of Contact Number
Deletion At Index | Press 1
Delete At Stack Format  | Press 2
Delete At Queue Format | Press 3
3
After Update
03002661270 Anas
03002661270 Taimoor
```

- **ConvertList2Binary FUNCTION:**

```
BinaryTreeNode convertList2Binary(BinaryTreeNode node)
{
    // queue to store the parent nodes
    Queue<BinaryTreeNode> q = new LinkedList<BinaryTreeNode>();
    // Base Case
    if (head == null)
    {
        node = null;
        return null;
    }
    // 1.) The first node is always the root node, and
    //     add it to the queue
    node = new BinaryTreeNode(head.data,head.name);
    q.add(node);
    //this root for taking the address of the first node of the tree
     root2 = q.peek();
    // advance the pointer to the next node
    head = head.next;
    // until the end of linked list is reached, do the
    // following steps
    while (head != null)
    {
        // 2.a) take the parent node from the q and
        //      remove it from q
        BinaryTreeNode parent = q.peek();
```

```
            BinaryTreeNode parent = q.peek();
            BinaryTreeNode pp = q.poll();
            // 2.c) take next two nodes from the linked list.
            // We will add them as children of the current
            // parent node in step 2.b. Push them into the
            // queue so that they will be parents to the
            // future nodes
            BinaryTreeNode left = null, right = null;
            left = new BinaryTreeNode(head.data,head.name);
            q.add(left);
            head = head.next;
            if (head != null)
            {
                right = new BinaryTreeNode(head.data,head.name);
                q.add(right);
                head = head.next;
            }
            // 2.b) assign the left and right children of
            //      parent
            parent.left = left;
            parent.right = right;
        }
        return node;
    }
```

- **SEARCH FUNCTION:**

```
public boolean Search(BinaryTreeNode Parent,String name)
{
         // it will work till parent !=null
        if (Parent != null)
        {
            //Here it is searching the Node
            if(Parent.name.equalsIgnoreCase(name))
            {
               System.out.println("Name: "+Parent.name+"\nPhone number: "+Parent.data);
               System.exit(0);
            }
            //Here we are using recursion
                 Search(Parent.left , name);
                 Search(Parent.right, name);
        }
        return false;

}
//Taking value from user for adding contact
```

```
Enter the contact name
Hasan
Name: Hasan
Phone number: 03002661270
```

**Print Function:**

```
public void print(listed list)
{
    Node currNode = list.head;
    //checking currNode
    while(currNode != null)
    {
        System.out.println(currNode.data + " " + currNode.name);
        currNode = currNode.next;
    }
}
```

```
03002661270 Hasan
03002661270 Anas
03002661270 Taimoor
```

```
void inorderTraversal(BinaryTreeNode node)
{
    //printing the traversing function
    if (node != null)
    {
        inorderTraversal(node.left);
        System.out.println(""+node.name + ":  "+node.data);
        inorderTraversal(node.right);
    }
}
```

**SORTED FUNCTION:**

```
Node Sort(Node list)
{
    Node i ,j;
    String temp=null;
    i=head;
    //this value checking with the other value of the linklist
    for (i=head; i!=null; i=i.next)
    {
        //this loop change the value for checking
        for (j =i.next; j!=null; j=j.next)
        {
            if (i.name.compareToIgnoreCase(j.name)<0)
            {
                temp=j.name;
                j.name=i.name;
                i.name=temp;

            }
        }
        System.out.println(""+i.name);
    }
    return list;
}
```

---

03002661270 Anas

03002661270 Hasan

03002661270 Taimoor

---