



9/2/2024

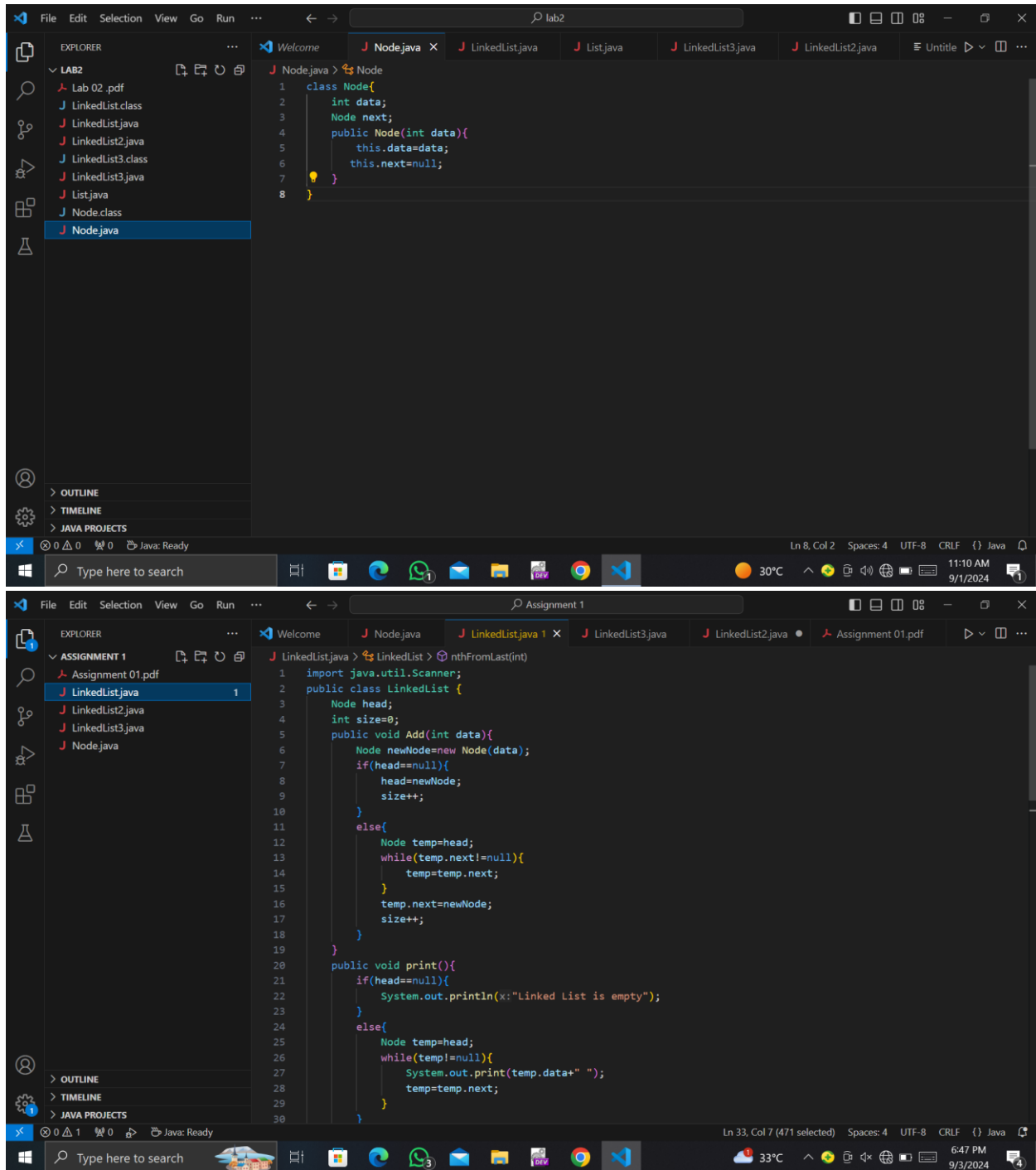
# Data Structures

## Assignment 1 Solutions

**Uzair Ali S/O Yar Muhammad Memon**  
**B-S CS III Section B**  
**023-23-0350**  
**To: Ma'am Marina Rajpoot**

## Task #1

## Solution:



The image displays two screenshots of a Visual Studio Code editor window, showing the implementation of a linked list in Java.

**Top Screenshot:** The editor is open to the `Node.java` file. The Explorer pane on the left shows a project named `LAB2` with files `LinkedList.class`, `LinkedList.java`, `LinkedList2.java`, `LinkedList3.class`, `LinkedList3.java`, `List.java`, `Node.class`, and `Node.java`. The `Node.java` file is selected. The code defines a `Node` class with an `int data` attribute and a `Node next` reference. The constructor `Node(int data)` initializes `this.data = data` and `this.next = null`.

```
1 class Node{
2     int data;
3     Node next;
4     public Node(int data){
5         this.data=data;
6         this.next=null;
7     }
8 }
```

**Bottom Screenshot:** The editor is open to the `LinkedList.java` file. The Explorer pane on the left shows a project named `ASSIGNMENT 1` with files `Assignment 01.pdf`, `LinkedList.java`, `LinkedList2.java`, `LinkedList3.java`, and `Node.java`. The `LinkedList.java` file is selected. The code defines a `LinkedList` class with a `Node head` attribute and an `int size` attribute. It includes methods `Add(int data)` and `print()`. The `Add` method creates a new node and adds it to the list. The `print` method prints the data of all nodes in the list.

```
1 import java.util.Scanner;
2 public class LinkedList {
3     Node head;
4     int size=0;
5     public void Add(int data){
6         Node newNode=new Node(data);
7         if(head==null){
8             head=newNode;
9             size++;
10        }
11        else{
12            Node temp=head;
13            while(temp.next!=null){
14                temp=temp.next;
15            }
16            temp.next=newNode;
17            size++;
18        }
19    }
20    public void print(){
21        if(head==null){
22            System.out.println(x:"Linked List is empty");
23        }
24        else{
25            Node temp=head;
26            while(temp!=null){
27                System.out.print(temp.data+" ");
28                temp=temp.next;
29            }
30        }
31    }
32 }
```

```
31 }
32
33 int nthFromLast(int n){
34     if(head==null){
35         System.out.println(x:"LinkedList is empty");
36         return 0;
37     }
38     else{
39         if(n==0 | n>size){
40             System.out.println(x:"Index out of bound");
41             return -1;
42         }
43         else{
44             Node temp=head;
45             while(size!=n) {
46                 temp=temp.next;
47                 size--;
48             }
49             return temp.data;
50         }
51     }
52 }
53
54 Run | Debug
55 public static void main(String[] args) {
56     Scanner sc=new Scanner(System.in);
57     LinkedList ll=new LinkedList();
58     int n;
59     ll.Add(data:10);
60     ll.Add(data:20);
61     ll.Add(data:30);
62     ll.Add(data:40);
63 }
```

```
60 ll.Add(data:50);
61 ll.Add(data:60);
62 ll.Add(data:70);
63 ll.Add(data:80);
64 ll.print();
65 System.out.println( );
66 System.out.print(s:"Enter the number of element to retrieve from the last : ");
67 n=sc.nextInt();
68 System.out.println( ll.nthFromLast(n));
69 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

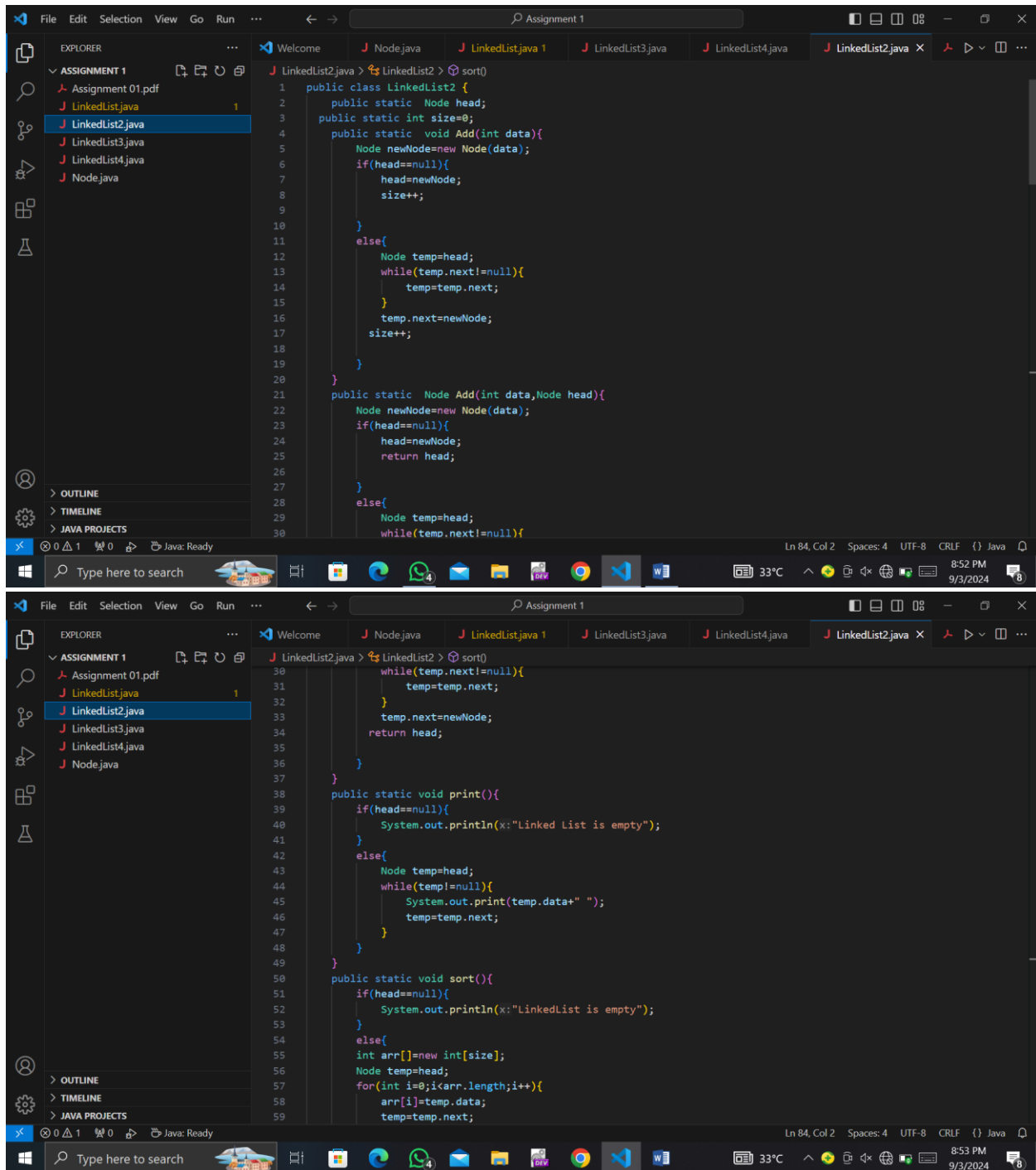
Windows PowerShell  
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS E:\bscs3\DSA lab\Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp ' 'C:\Users\BOSS\AppData\Roaming\Code\User\workspaceStorage\5d876ddfbe4f882b3f26564f28be6ed1\redhat.java\jdt_ws\Assignment_1_889981a5\bin' 'LinkedList'
10 20 30 40 50 60 70 80
Enter the number of element to retrieve from the last : 3
60
PS E:\bscs3\DSA lab\Assignment 1>
```

## Task#2

## Solution:



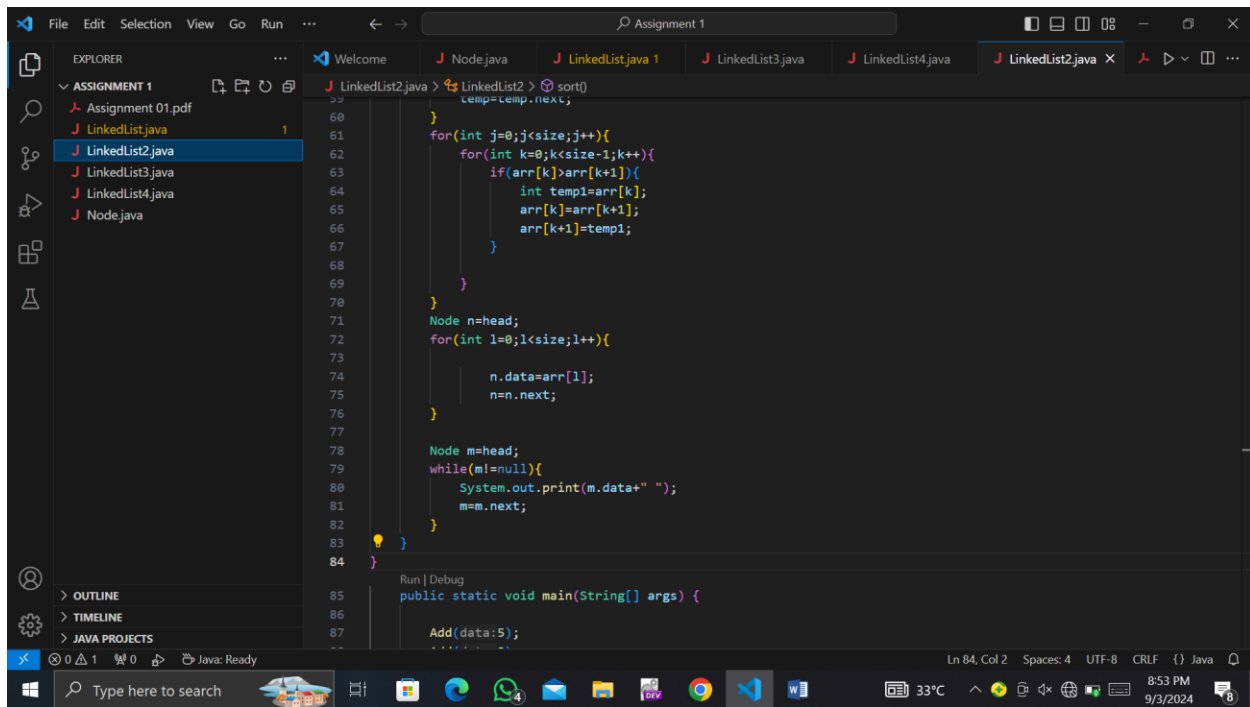
The image shows two screenshots of a code editor (VS Code) displaying a Java program for a linked list. The Explorer panel on the left shows a project named 'ASSIGNMENT 1' with files: 'Assignment 01.pdf', 'LinkedList.java', 'LinkedList2.java', 'LinkedList3.java', 'LinkedList4.java', and 'Node.java'. The 'LinkedList2.java' file is selected and open in the editor.

The code in the first screenshot (lines 1-30) defines a `LinkedList2` class with a static `head` pointer and a `size` variable. It includes an `Add(int data)` method that adds a new node to the end of the list and a static `Add(int data, Node head)` method that returns the new head.

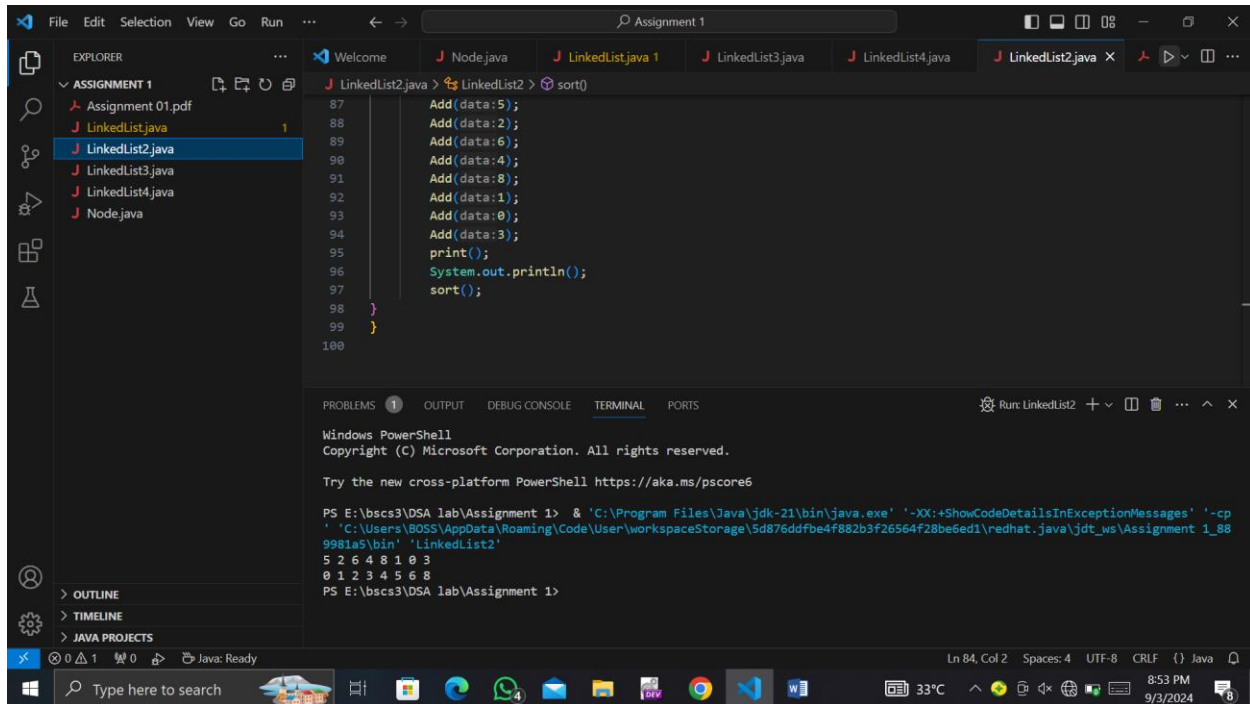
The code in the second screenshot (lines 30-59) continues the implementation with a `print()` method to display the list contents and a `sort()` method that sorts the list using an array-based approach.

```
1 public class LinkedList2 {
2     public static Node head;
3     public static int size=0;
4     public static void Add(int data){
5         Node newNode=new Node(data);
6         if(head==null){
7             head=newNode;
8             size++;
9         }
10    }
11    else{
12        Node temp=head;
13        while(temp.next!=null){
14            temp=temp.next;
15        }
16        temp.next=newNode;
17        size++;
18    }
19 }
20
21 public static Node Add(int data,Node head){
22     Node newNode=new Node(data);
23     if(head==null){
24         head=newNode;
25         return head;
26     }
27     else{
28         Node temp=head;
29         while(temp.next!=null){
30
```

```
30         while(temp.next!=null){
31             temp=temp.next;
32         }
33         temp.next=newNode;
34         return head;
35     }
36 }
37
38 public static void print(){
39     if(head==null){
40         System.out.println(x:"Linked List is empty");
41     }
42     else{
43         Node temp=head;
44         while(temp!=null){
45             System.out.print(temp.data+" ");
46             temp=temp.next;
47         }
48     }
49 }
50
51 public static void sort(){
52     if(head==null){
53         System.out.println(x:"LinkedList is empty");
54     }
55     else{
56         int arr[]=new int[size];
57         Node temp=head;
58         for(int i=0;i<arr.length;i++){
59             arr[i]=temp.data;
60             temp=temp.next;
61         }
62         Arrays.sort(arr);
63         temp=head;
64         for(int i=0;i<arr.length;i++){
65             temp.data=arr[i];
66             temp=temp.next;
67         }
68     }
69 }
```



```
File Edit Selection View Go Run ...
Assignment 1
EXPLORER
ASSIGNMENT 1
  Assignment 01.pdf
  LinkedList.java
  LinkedList2.java
  LinkedList3.java
  LinkedList4.java
  Node.java
OUTLINE
TIMELINE
JAVA PROJECTS
J LinkedList2.java > LinkedList2 > sort()
temp=temp.next;
}
for(int j=0;j<size;j++){
  for(int k=0;k<size-1;k++){
    if(arr[k]>arr[k+1]){
      int temp1=arr[k];
      arr[k]=arr[k+1];
      arr[k+1]=temp1;
    }
  }
}
Node n=head;
for(int l=0;l<size;l++){
  n.data=arr[l];
  n=n.next;
}
Node m=head;
while(m!=null){
  System.out.print(m.data+" ");
  m=m.next;
}
}
Run | Debug
public static void main(String[] args) {
  Add(data:5);
  ...
}
Ln 84, Col 2 Spaces: 4 UTF-8 CRLF {} Java
8:53 PM 9/3/2024
```



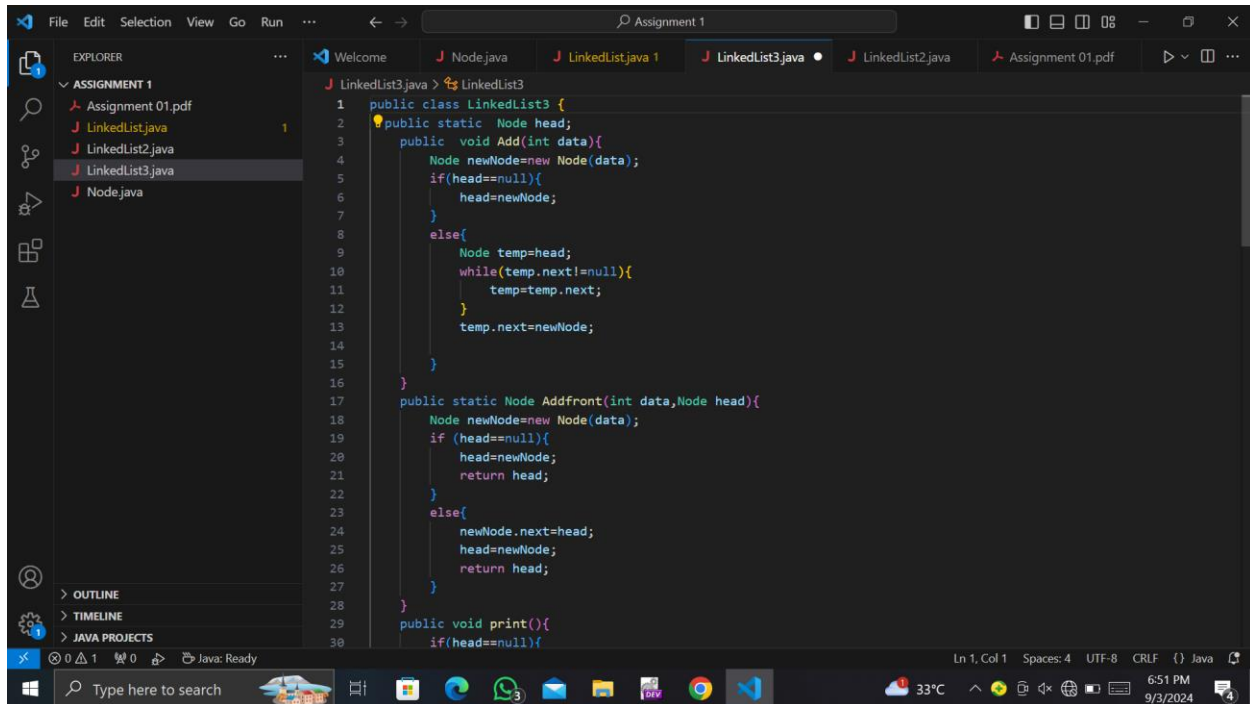
```
File Edit Selection View Go Run ...
Assignment 1
EXPLORER
ASSIGNMENT 1
  Assignment 01.pdf
  LinkedList.java
  LinkedList2.java
  LinkedList3.java
  LinkedList4.java
  Node.java
OUTLINE
TIMELINE
JAVA PROJECTS
J LinkedList2.java > LinkedList2 > sort()
Add(data:5);
Add(data:2);
Add(data:6);
Add(data:4);
Add(data:8);
Add(data:1);
Add(data:0);
Add(data:3);
print();
System.out.println();
sort();
}
}
100
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Run: LinkedList2
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS E:\bscs3\DSA lab\Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp 'C:\Users\BOSS\AppData\Roaming\Code\User\workspaceStorage\5d876ddf4f882b3f26564f28be6ed1\redhat.java\jdt_ws\Assignment_1_889981a5\bin' 'LinkedList2'
5 2 6 4 8 1 0 3
0 1 2 3 4 5 6 8
PS E:\bscs3\DSA lab\Assignment 1>
Ln 84, Col 2 Spaces: 4 UTF-8 CRLF {} Java
8:53 PM 9/3/2024
```

## Task#3

## Solution:



The screenshot shows a Visual Studio Code editor with a project named 'ASSIGNMENT 1'. The Explorer panel on the left lists the files: 'Assignment 01.pdf', 'LinkedList.java', 'LinkedList2.java', 'LinkedList3.java', and 'Node.java'. The 'LinkedList3.java' file is selected and its code is displayed in the main editor. The code defines a 'LinkedList3' class with a static 'Node head' and two methods: 'Add(int data)' and 'Addfront(int data, Node head)'. The 'Add' method creates a new node and adds it to the end of the list. The 'Addfront' method creates a new node and adds it to the beginning of the list. A 'print()' method is also present but not fully shown.

```
1 public class LinkedList3 {
2     public static Node head;
3     public void Add(int data){
4         Node newNode=new Node(data);
5         if(head==null){
6             head=newNode;
7         }
8     }
9     else{
10        Node temp=head;
11        while(temp.next!=null){
12            temp=temp.next;
13        }
14        temp.next=newNode;
15    }
16 }
17 public static Node Addfront(int data,Node head){
18     Node newNode=new Node(data);
19     if (head==null){
20         head=newNode;
21         return head;
22     }
23     else{
24         newNode.next=head;
25         head=newNode;
26         return head;
27     }
28 }
29 public void print(){
30     if(head==null){
```

The status bar at the bottom indicates the current position is 'Ln 1, Col 1' with 'Spaces: 4', 'UTF-8' encoding, and 'CRLF' line endings. The system tray shows the date and time as '6:51 PM 9/3/2024'.

```
30 if(head==null){
31     System.out.println(x:"Linked List is empty");
32 }
33 else{
34     Node temp=head;
35     while(temp!=null){
36         System.out.print(temp.data+" ");
37         temp=temp.next;
38     }
39 }
40
41 public static void reverse(){
42     if( head==null){
43         System.out.println(x:"Linked List is empty");
44     }
45     else{
46         Node head2=null;
47         Node temp=head;
48         while(temp!=null){
49             head2=Addfront(temp.data,head2);
50             temp=temp.next;
51         }
52         Node n=head2;
53         while(n!=null){
54             System.out.print(n.data+" ");
55             n=n.next;
56         }
57     }
58 }
59 }
```

```
60 Run | Debug
61 public static void main(String[] args) {
62     LinkedList3 ll=new LinkedList3();
63     ll.Add(data:10);
64     ll.Add(data:20);
65     ll.Add(data:30);
66     ll.Add(data:40);
67     ll.Add(data:50);
68     ll.Add(data:60);
69     ll.Add(data:70);
70     ll.Add(data:80);
71     ll.print();
72     System.out.println();
73     reverse();
74 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Run: LinkedList3

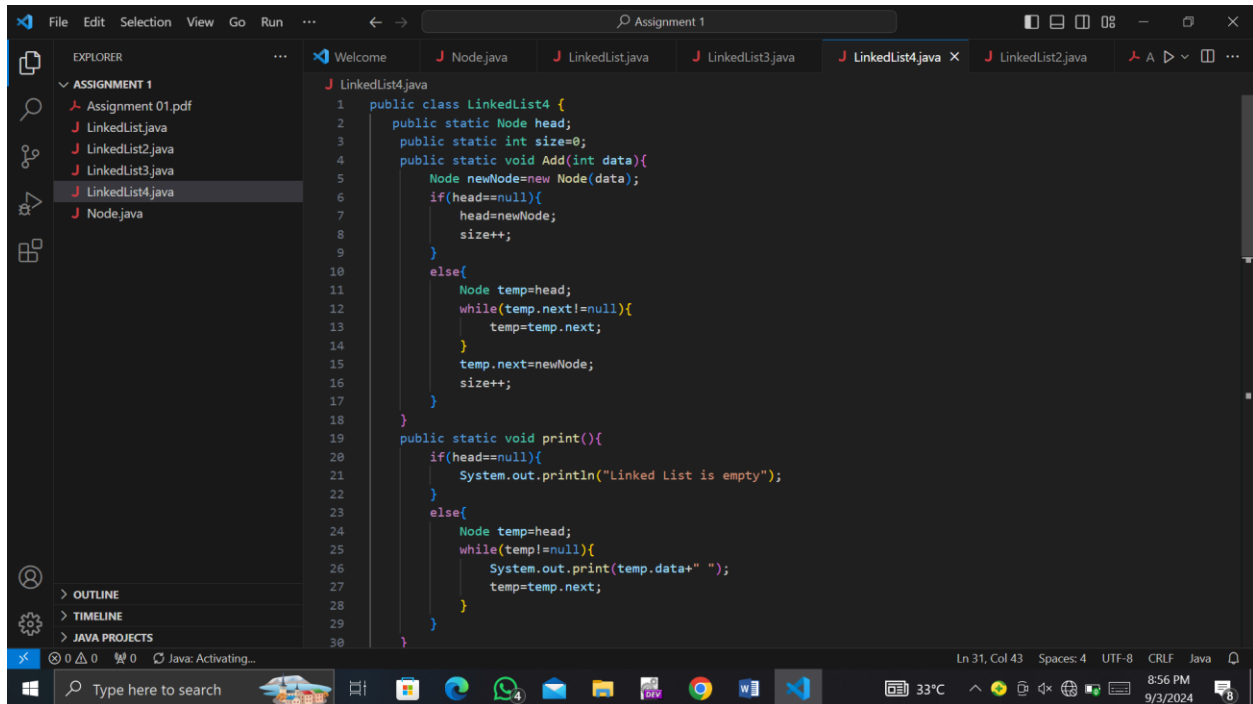
Windows PowerShell  
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS E:\bscs3\DSA lab\Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp 'C:\Users\BOSS\AppData\Roaming\Code\User\workspaceStorage\5d876ddfbe4f882b3f26564f28be6ed1\redhat.java\jdt_ws\Assignment_1_889981a5\bin' 'LinkedList3'
10 20 30 40 50 60 70 80
80 70 60 50 40 30 20 10
PS E:\bscs3\DSA lab\Assignment 1>
```

## Task#4

## Solution:



The screenshot shows an IDE with the following components:

- Explorer:** A file tree on the left showing a project named "ASSIGNMENT 1" containing files: "Assignment 01.pdf", "LinkedList.java", "LinkedList2.java", "LinkedList3.java", "LinkedList4.java" (selected), and "Node.java".
- Editor:** The main window displays the code for "LinkedList4.java". The code implements a linked list with an "Add" method and a "print" method.
- Output/Console:** The bottom of the editor shows the "print" method's logic, which iterates through the list and prints each node's data.
- Taskbar:** The Windows taskbar at the bottom shows the system clock as 8:56 PM on 9/3/2024, along with various application icons and a temperature display of 33°C.

```
1 public class LinkedList4 {  
2     public static Node head;  
3     public static int size=0;  
4     public static void Add(int data){  
5         Node newNode=new Node(data);  
6         if(head==null){  
7             head=newNode;  
8             size++;  
9         }  
10        else{  
11            Node temp=head;  
12            while(temp.next!=null){  
13                temp=temp.next;  
14            }  
15            temp.next=newNode;  
16            size++;  
17        }  
18    }  
19    public static void print(){  
20        if(head==null){  
21            System.out.println("Linked List is empty");  
22        }  
23        else{  
24            Node temp=head;  
25            while(temp!=null){  
26                System.out.print(temp.data+" ");  
27                temp=temp.next;  
28            }  
29        }  
30    }
```



This screenshot shows the Visual Studio Code editor with the file `LinkedList4.java` open. The code implements a method `removeDuplicates()` that traverses a linked list and removes duplicate nodes. The `main` method calls `Add(5)` and `Add(4)`.

```
31 public static void removeDuplicates() {
32     if (head == null) {
33         System.out.println("LIST_EMPTY");
34         return;
35     }
36
37     Node temp1 = head;
38     while (temp1 != null) {
39         Node temp2 = temp1;
40         while (temp2.next != null) {
41             if (temp2.next.data == temp1.data) {
42                 // duplicate removed
43                 temp2.next = temp2.next.next;
44             } else {
45                 temp2 = temp2.next;
46             }
47         }
48         temp1 = temp1.next;
49     }
50
51     Node n = head;
52     while (n != null) {
53         System.out.print(n.data + " ");
54         n = n.next;
55     }
56 }
57
58 public static void main(String[] args) {
59     Add(5);
60     Add(4);
61 }
```

This screenshot shows the same Visual Studio Code editor with the `LinkedList4.java` file. The `main` method has been updated to include `print()` and `removeDuplicates()` calls. The terminal window at the bottom shows the output of the program, which is the sequence of numbers 5 4 3 2 1 4 5 2, followed by the result after removing duplicates: 5 4 3 2 1.

```
60 Add(data:4);
61 Add(data:3);
62 Add(data:2);
63 Add(data:1);
64 Add(data:4);
65 Add(data:5);
66 Add(data:2);
67 print();
68 System.out.println();
69 removeDuplicates();
70
71 }
72
73
```

Windows PowerShell  
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS E:\bscs3\DSA lab\Assignment 1> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp 'C:\Users\BOSS\AppData\Roaming\Code\User\workspaceStorage\5d876ddf4f882b3f26564f28be6ed1\redhat.java\jdt_ws\Assignment_1_889981a5\bin' 'LinkedList4'
5 4 3 2 1 4 5 2
5 4 3 2 1
PS E:\bscs3\DSA lab\Assignment 1>
```

**THE END**