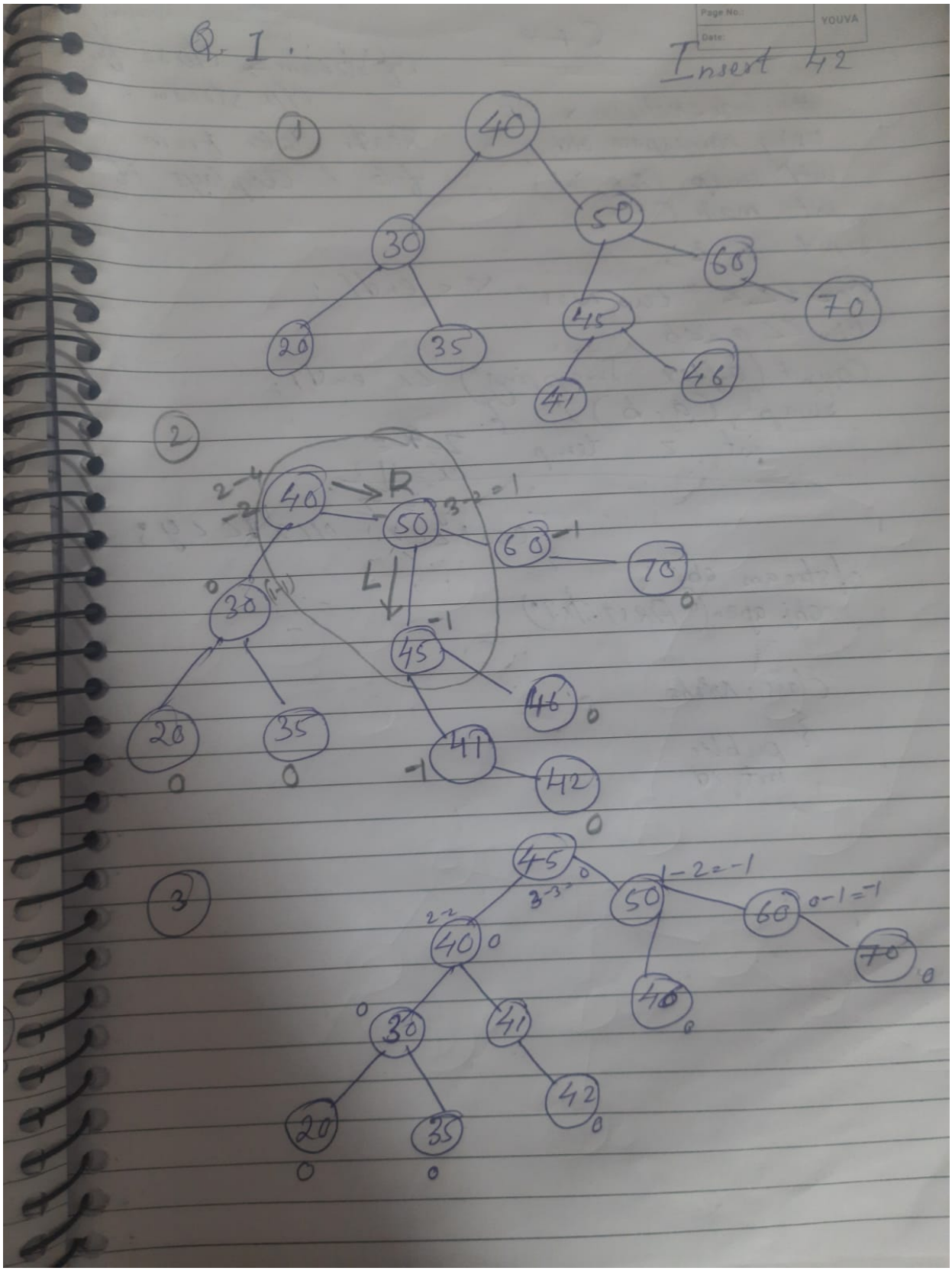
LAB ASSIGNMENT DSA -5

Q1. Draw an AVL Tree and Insert node 42

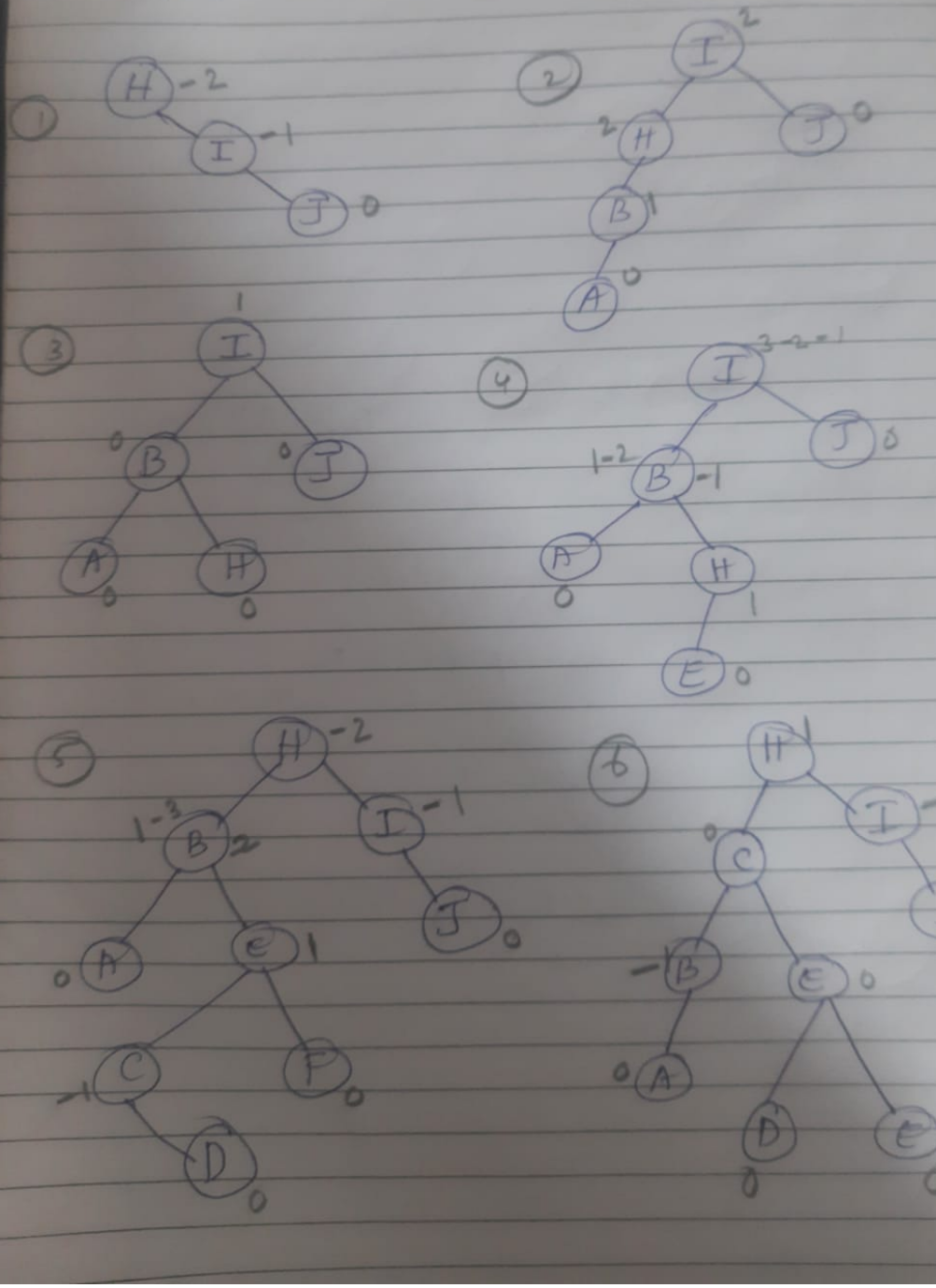
Q. 1.

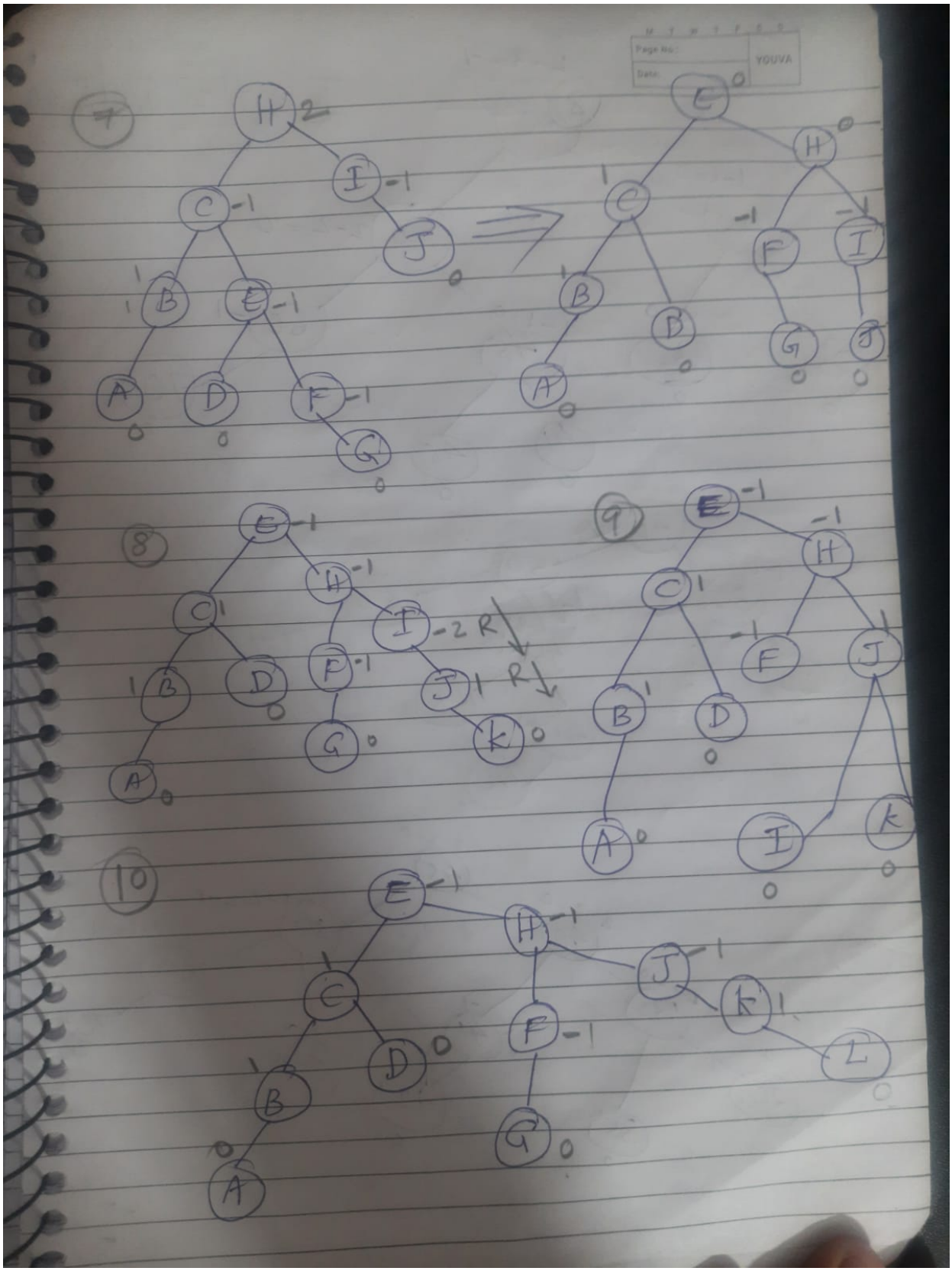
Insert 42


1
40
30
50
20
35
45
60
70
41
46
2
40
R
50
60
70
30
L
45
20
35
46
41
42
3
45
40
50
60
70
30
41
46
42
20
35

Q2 .Construct an AVL tree having the following elements H, I, J, B, A, E, C, F, D, G, K, L and the final AVL Tree should be Balanced.

Q. 2. H, I, J, B, A, E, C, F, D, G, K, L




7
H 2
C -1
I -1
B 1
E -1
J 0
A 0
D 0
F -1
G 0
C 0
H 0
C 1
F -1
I -1
B 1
D 0
G 0
J 0
A 0
8
E -1
H -1
C 1
I -2 R
B 1
D 0
F -1
J 1 R
G 0
K 0
A 0
9
E -1
H -1
C 1
F -1
J 1
B 1
D 0
A 0
I 0
K 0
10
E -1
H -1
C 1
J -1
F -1
K 1
D 0
B 1
L 0
G 0
A 0

Q3. Write a Java program to Create an array and insert the elements like
{ 15 , 32 , 24, 67 ,49,10}
1. Using linear Search Find the position of 67
2. Insert an element 50 and display the message “Element not Found”

```java
public class LinearSearch {
    public static int search(int arr[], int element) {
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == element) {
                return i;
            }
        }
        return -1;
    }

    public static int[] insertElement(int arr[], int element) {
        int newArray[] = new int[arr.length + 1];
        for (int i = 0; i < arr.length; i++) {
            newArray[i] = arr[i];
        }
        newArray[arr.length] = element;
        return newArray;
    }

    Run | Debug
    public static void main(String[] args) {
        int arr[] = {15, 32, 24, 67, 49, 10};
        int element = 67;

        int position = search(arr, element);

        if (position != -1) {
            System.out.println("Position of " + element + " is " + position);
        } else {
            System.out.println("Element not Found");

            // Insert the element 50
            arr = insertElement(arr, element:50);
            System.out.println("Element 50 inserted at the end");
        }
    }
}
```

```
nisha@nisha-Cloud:/media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/
DSA/day5/LIneraSearch$  cd /media/sf_Vertual_Box_Share/Nisha_Ubu
ntu/Cdac/DSA/day5/LIneraSearch ; /usr/bin/env /usr/lib/jvm/java-
8-openjdk-amd64/jre/bin/java -cp /home/nisha/.config/Code/User/w
orkspaceStorage/aa9ab66a186875fa7724195f4aa52b98/redhat.java/jdt
_ws/LIneraSearch_8b34c14c/bin LinearSearch
Position of 67 is 3
```

Q4. Create an array and insert the elements in sorted order
Using Binary search Find the particular position of the element, if the element was not found the show the message “Element Not Found”.

```java
public class BinarySearch {
    public int search(int arr[], int start, int end, int element) {
        if (start == end) {
            if (arr[start] == element)
                return start;
            else
                return -1;
        }

        int mid = (start + end) / 2;

        if (element == arr[mid])
            return mid;

        if (element > arr[mid])
            return search(arr, (mid + 1), end, element);
        else
            return search(arr, start, (mid - 1), element);
    }

    Run | Debug
    public static void main(String[] args) {
        BinarySearch obj = new BinarySearch();
        int arr[] = {15, 26, 37, 38, 49, 103}; // Sorted array in ascending order
        int n, element;
        n = arr.length - 1;
        element = 33;

        int result = obj.search(arr, start:0, n, element);

        if (result != -1) {
            System.out.println("Position of " + element + " is " + result);
        } else {
            System.out.println("Element Not Found");
        }
    }
}
```

```
nisha@nisha-Cloud:/media/sf_Vertual_Box_Shar
e/Nisha_Ubuntu/Cdac/DSA/day5$  /usr/bin/env
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/ja
va -cp /home/nisha/.config/Code/User/worksp
ceStorage/bfd68a0f82a9a32fab9ff31caf66af40/r
edhat.java/jdt_ws/day5_694147ce/bin BinarySe
arch
Element Not Found
```

Q5.Create an array & insert the elements like 5 1 6 2 4 3 7 sort the array in Assenting

order using Bubble sort and display the result

```java
public class BubbleSort{
    public static void sort(int arr[]) {
        for (int i = 0; i < arr.length - 1; i++) {
            int flag = 0;
            for (int j = 0; j < arr.length - 1 - i; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                    flag = 1;
                }
            }
            if (flag == 0) {
                break;
            }
        }
    }

    Run | Debug
    public static void main(String[] args) {
        int[] arr = {5, 1, 6, 2, 4, 3, 7};
        System.out.print("Array in Given Order: ");
        for (int i : arr) {
            System.out.print(i + " ");
        }
        System.out.println();
        sort(arr);

        System.out.println("Sorted Array in Ascending Order:")
        for (int i : arr) {
            System.out.print(i + " ");
        }
        System.out.println();
    }
}
```

```
DSA/day5/BubbleSort ; /usr/bin/env /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java -cp /hor
8e542a9744036cf3789dd26beac541/redhat.java/jdt_ws/BubbleSort_407a114b/bin BubbleSort
Array in Given Order: 5 1 6 2 4 3 7
Sorted Array in Ascending Order:
1 2 3 4 5 6 7
nisha@nisha-Cloud:/media/sf_Vertual_Box_Share/Nisha_Ubuntu/Cdac/DSA/day5/BubbleSort$
```