

DAILY ONLINE ACTIVITIES SUMMARY

Date:	27-05-2020	Name:	Ashika
Sem & Sec	6 A	USN:	4AL17CS016
Online Test Summary			
Subject	SSCD		
Max. Marks	30	Score	14
Certification Course Summary			
Course	ETHICAL HACKING		
Certificate Provider	GREAT LEARNING	Duration	6 HOUR
Coding Challenges			

Problem Statement:

1. Given an array `arr[]` of the positive integers of size `N`, the task is to find the largest element on the left side of each index which is smaller than the element present at that index. Note: If no such element is found then print `-1`.

2. 2. Write a Java program to implement Binary Tree using the Linked List

In this program, we need to create the binary tree by inserting nodes and displaying nodes in in-order fashion. A typical binary tree can be represented as follows:

Java program to implement Binary Tree using the Linked List

In the binary tree, each node can have at most two children. Each node can have zero, one or two children. Each node in the binary tree contains the following information:

Java program to implement Binary Tree using the Linked List
Data that represents value stored in the node.

Left that represents the pointer to the left child.

Right that represents the pointer to the right child.

Algorithm

Define Node class which has three attributes namely: data left and right. Here, left represents the left child of the node and right represents the right child of the node.

When a node is created, data will pass to data attribute of the node and both left and right will be set to null.

Define another class which has an attribute root.

Root represents the root node of the tree and initialize it to null.

a. `insert()` will add a new node to the tree:

It checks whether the root is null, which means the tree is empty. It will add the new node as root.

Else, it will add root to the queue.

The variable node represents the current node.

First, it checks whether a node has a left and right child. If yes, it will add both nodes to queue.

If the left child is not present, it will add the new node as the left child.

If the left is present, then it will add the new node as the right child.

a. `Inorder()` will display nodes of the tree in inorder fashion.

3. 3. In Bubble sort, each pass consists of comparison each element in the file with its successor (i.e. `x[i]` with `x[i+1]`) and interchanging two elements if they are not in the proper order. The array may be sorted in any pass. If the array is sorted, then remaining passes should be skipped off. Write a C Program to sort an array of integers in ascending order and display the sorted array and

Status: executed	
Uploaded the report in Github	yes
If yes Repository name	https://github.com/ASHIKA-05/27-05-2020
Uploaded the report in slack	yes

SUBJECT: SSCD

The screenshot displays a 'Results' page with three test result cards. Each card shows a green checkmark icon, the test name, the round type, and the score. The scores are: Test 1 (1/11), Test 2 (3/9), and Test 3 (10/10). The Windows taskbar at the bottom shows the search bar, task view, and several pinned applications including Edge, Mail, and Chrome. The system clock indicates the time is 09:54 on 2020-05-27.

Test	Round	Score
Test 1	MCQ	1 / 11
Test 2	Problem Round 1	3 / 9
Test 3	Problem Round 2	10 / 10

CERTIFICATION COURSE

- Quiz

greatlearning
Learning for Life

```
import jakar.seem.diva;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class IngActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_log);
    }

    public void checkOut(View view) {
        EditText ctext = (EditText) findViewById(R.id.ctext);
        try {
            // Assuming we do some HTTP requests credit card validation and processing
            //everything seems fine and then we hit some unforeseen error
            process(ctext.getText().toString());
        } catch (RuntimeException re) {
            Log.e("diva-log", "Error while processing transaction with credit card: " + ctext.getText().toString());
            Toast.makeText(this, "An error occurred. Please try again later.", Toast.LENGTH_SHORT).show();
        }
    }
}
```



Apps



Gmail



YouTube



Maps



Content

Learning Videos



Career and Growth Ladder in
Ethical Hacking



Domains and Process
Implementation under Ethical
Hacking



Ethical Hacking in Network
Architecture-Demonstration



Ethical Hacking in Web
Applications-Demonstration



Ethical Hacking on Mobile
Platforms-Demonstration



What is Ethical Hacking

Quiz



ONLINE CODEING

1. Given an array `arr[]` of the positive integers of size `N`, the task is to find the largest element on the left side of each index which is smaller than the element present at that index. Note: If no such element is found then print `-1`.

```
import java.util.*;

class Main{

static void findMaximumBefore(int arr[], int n){

    for (int i = 0; i < n; i++) {

        int currAns = -1;

        for (int j = i - 1; j >= 0; j--) {

            if (arr[j] > currAns &&

                arr[j] < arr[i]) {

                    currAns = arr[j];

            }

        }

        System.out.print(currAns+ " ");

    }

}

public static void main(String[] args)

{

    int arr[] = { 4,7,6,8,5};

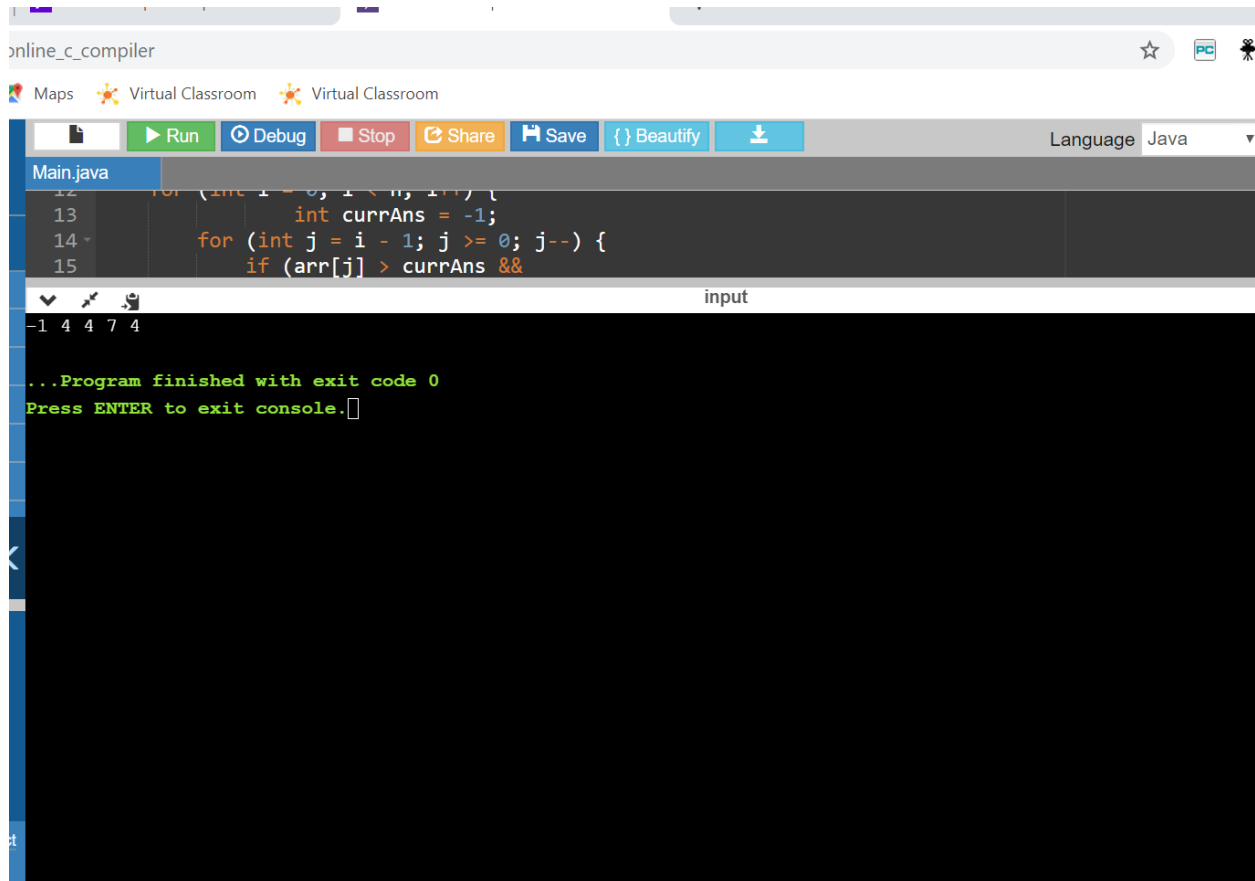
    int n = arr.length;

    findMaximumBefore(arr, n);

}

}
```

Output:



The screenshot shows an online Java compiler interface. The top bar includes a search bar with 'online_c_compiler', a star icon, a PC icon, and a user icon. Below the bar are links for 'Maps', 'Virtual Classroom', and 'Virtual Classroom'. The main toolbar contains buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', 'Beautify', and a download icon. The 'Language' dropdown is set to 'Java'. The code editor shows 'Main.java' with the following code:

```
12 for (int i = 0; i < n; i++) {  
13     int currAns = -1;  
14     for (int j = i - 1; j >= 0; j--) {  
15         if (arr[j] > currAns &&
```

The output console shows the input '-1 4 4 7 4' and the message '...Program finished with exit code 0'. It also prompts 'Press ENTER to exit console.'.

2. Write a Java program to implement Binary Tree using the Linked List

In this program, we need to create the binary tree by inserting nodes and displaying nodes in in-order fashion. A typical binary tree can be represented as follows:

Java program to implement Binary Tree using the Linked List

In the binary tree, each node can have at most two children. Each node can have zero, one or two children. Each node in the binary tree contains the following information:

Java program to implement Binary Tree using the Linked List

Data that represents value stored in the node.

Left that represents the pointer to the left child.

Right that represents the pointer to the right child.

Algorithm

Define Node class which has three attributes namely: data left and right. Here, left represents the left child of the node and right represents the right child of the node.

When a node is created, data will pass to data attribute of the node and both left and right will be set to null.

Define another class which has an attribute root.

Root represents the root node of the tree and initialize it to null.

a. insert() will add a new node to the tree:

It checks whether the root is null, which means the tree is empty. It will add the new node as root.

Else, it will add root to the queue.

The variable node represents the current node.

First, it checks whether a node has a left and right child. If yes, it will add both nodes to queue.

If the left child is not present, it will add the new node as the left child.

If the left is present, then it will add the new node as the right child.

a. Inorder() will display nodes of the tree in inorder fashion.

It traverses the entire tree then prints out left child followed by root then followed by the right child.

```
public class BinarySearchTree {
```

```
    public static class Node{
```

```
        int data;
```

```
        Node left;
```

```
        Node right;
```

```
        public Node(int data){
```

```
            this.data = data;
```

```
            this.left = null;
```

```
            this.right = null;
```

```
        }
```

```
    }
```



```

public Node root;

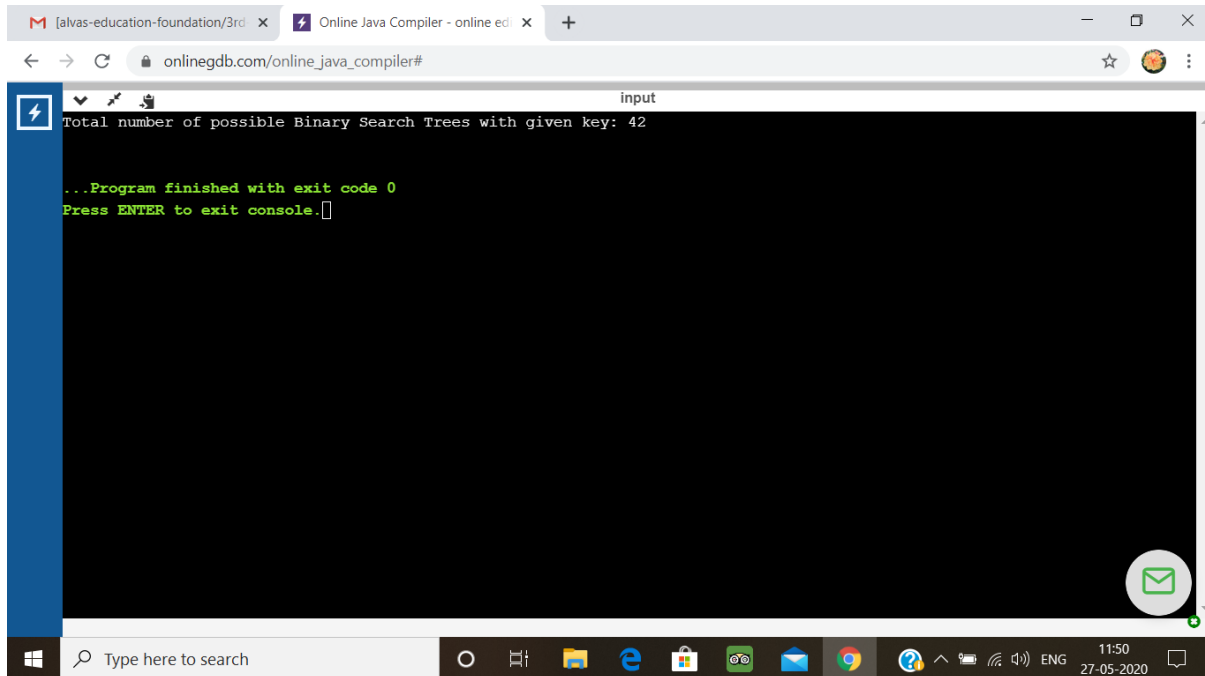
public BinarySearchTree(){
    root = null;
}

public int factorial(int num) {
    int fact = 1;
    if(num == 0)
        return 1;
    else {
        while(num > 1) {
            fact = fact * num;
            num--;
        }
        return fact;
    }
}

public int numOfBST(int key) {
    int catalanNumber = factorial(2 * key)/(factorial(key + 1) * factorial(key));
    return catalanNumber;
}

public static void main(String[] args) {
    BinarySearchTree bt = new BinarySearchTree();
    System.out.println("Total number of possible Binary Search Trees with given key: " + bt.numOfBST(5));
}
}

```



3. In Bubble sort, each pass consists of comparison each element in the file with its successor (i.e. $x[i]$ with $x[i+1]$) and interchanging two elements if they are not in the proper order. The array may be sorted in any pass. If the array is sorted, then remaining passes should be skipped off. Write a C Program to sort an array of integers in ascending order and display the sorted array and Number of passes performed for sorting

```
#include <stdio.h>
```

```
void swap(int *xp, int *yp)
```

```
{
```

```
    int temp = *xp;
```

```
    *xp = *yp;
```

```
    *yp = temp;
```

```
}
```

```
void bubbleSort(int arr[], int n)
```

```
{
```

```

int i, j;

for (i = 0; i < n-1; i++)
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}

void printArray(int arr[], int size)
{
    int i;
    for (i=0; i < size; i++)
        printf("%d ", arr[i]);
    printf("\n");
}

int main()
{
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    bubbleSort(arr, n);
    printf("Sorted array: \n");
    printArray(arr, n);
    return 0;
}

```

Output:

SPONSOR

Red Hat — Scale up, retract, hold steady: Hybrid cloud gives you flexibility to do what times demand.

Run

Debug

Stop

Share

Save

{ } Beautify

Language C

main.c

23

24

swap(arr[j], arr[j+1]);

input

Sorted array:

11 12 22 25 34 64 90

...Program finished with exit code 0

Press ENTER to exit console.