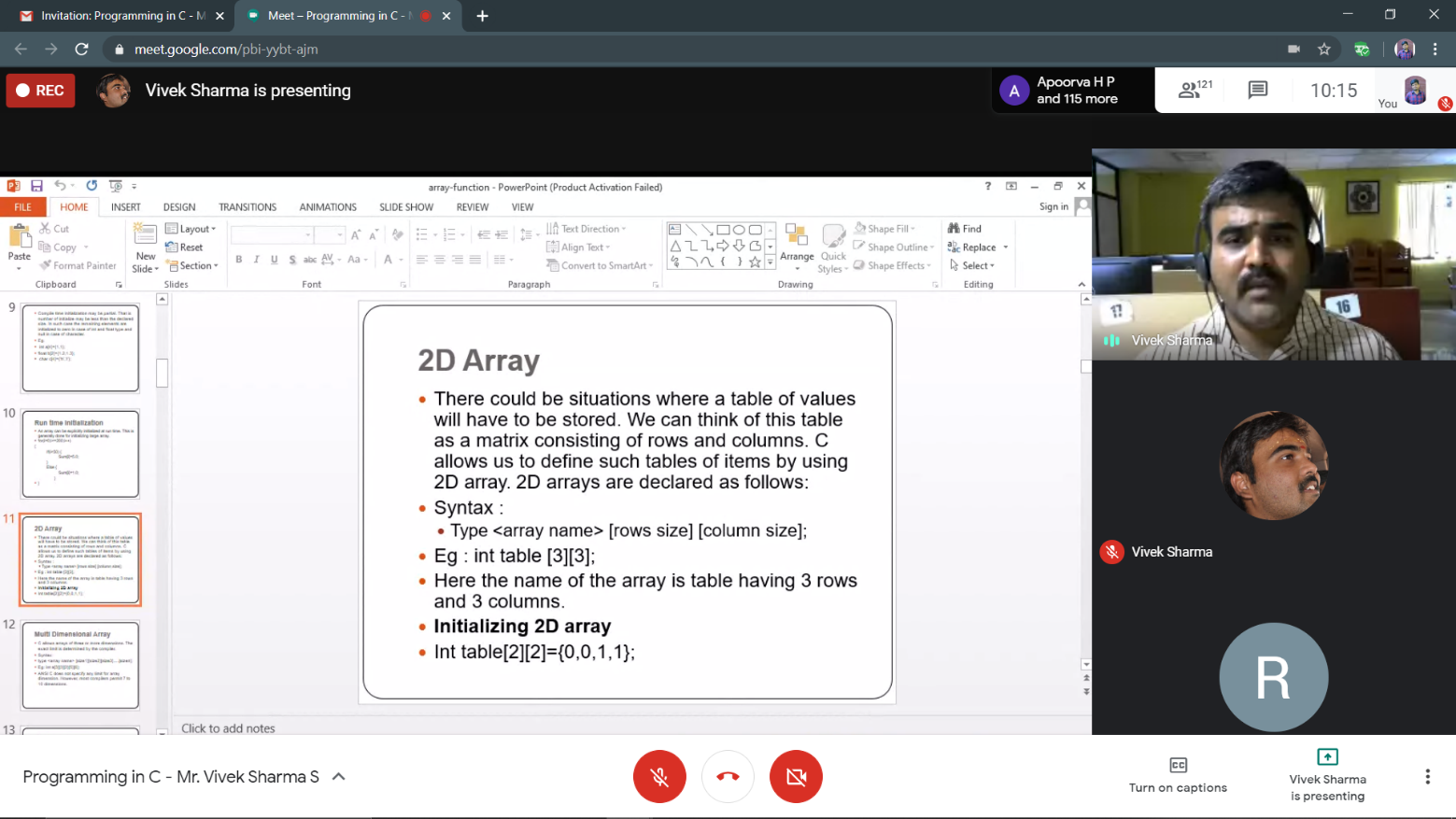
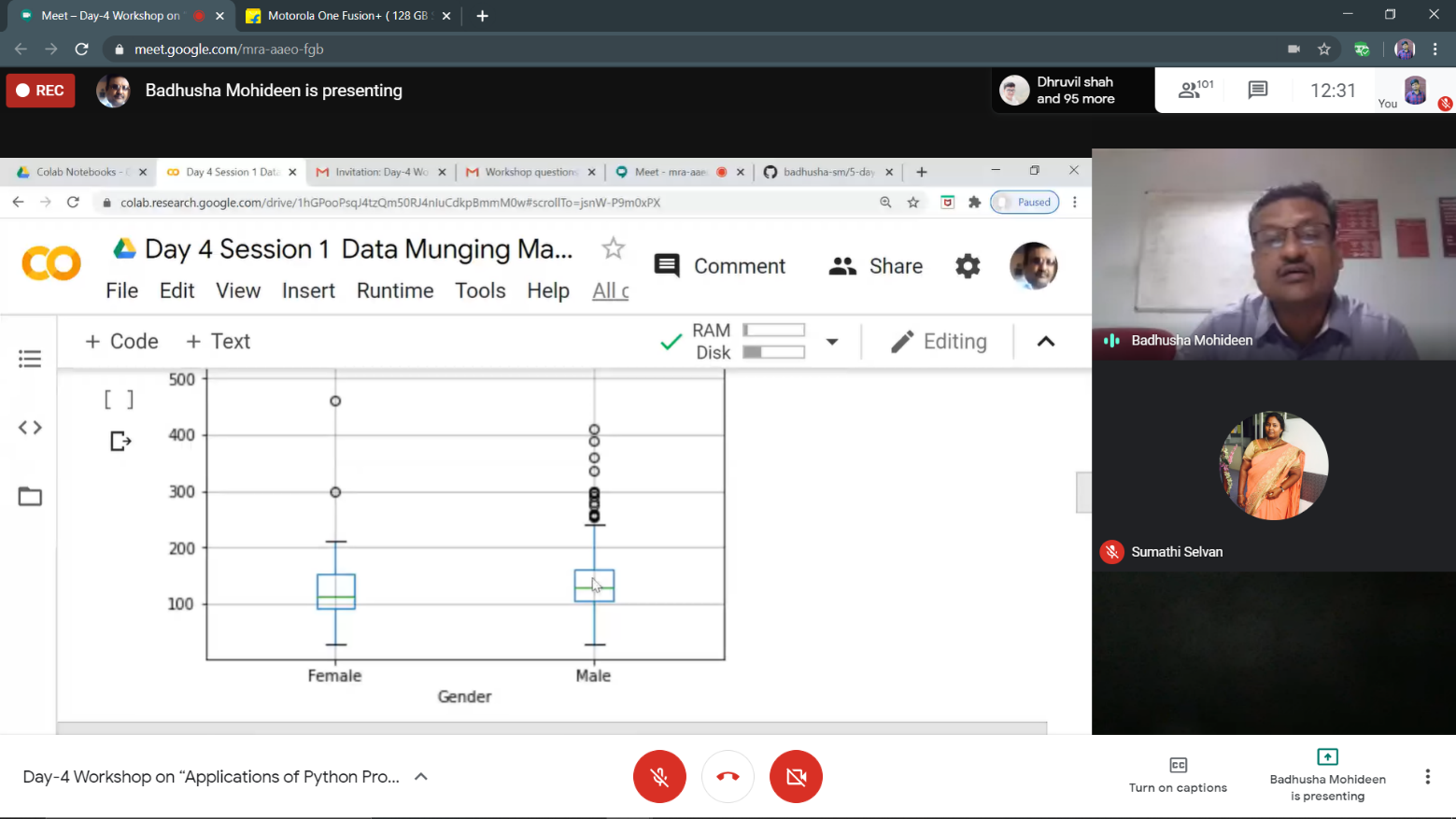
**“DAILY ONLINE ACTIVITIES SUMMARY”**\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | **18-06-2020** | | | **Name:** | **TANOJ M** |
| **Sem & Sec** | **6th - A** | | | **USN:** | **4AL16CS113** |
| **PRE-PLACEMENT TRAINING** | | | | | |
| **Subject** | **C-Programing(Arrays) and Python workshop(day-4)** | | | | |
| **Max. Marks** | **--** | **Score** | | | **--** |
| **C-Programing**  **Python workshop** | 9:15 to 11:00 Mr. Vivek Sharma  11:00 to 1:40 Dr. Mohideen Badhusha | | | | |
| **Coding Challenges** | | | | | |
| **Problem Statemen:**   1. **Java program to check BST or NOT** 2. **C program to get n magic number** | | | | | |
| **Status: executed** | | | | | |
| **Uploaded the report in Github** | | | **yes** | | |
| **If yes Repository name** | | | [**https://github.com/Tanoj8296/DAILY-STATUS**](https://github.com/Tanoj8296/DAILY-STATUS) | | |
| **Uploaded the report in slack** | | | **Yes** | | |

Training:





Python workshop :

**Assessments:**

**Uploaded in github account and respective links are provided.**

[**https://github.com/Tanoj8296/PRE-PLACEMENT-TRAINING**](https://github.com/Tanoj8296/PRE-PLACEMENT-TRAINING)

**ONLINE CODDING CHALLENGES:**

 Write a C Program to generate first N Magic Numbers.

#include<stdio.h>

int nthMagicNo(int n)

{

int pow = 1, answer = 0;

while (n)

{

pow = pow\*5;

if (n & 1)

answer += pow;

printf("%d ",pow);

n >>= 1;

}

}

int main()

{

printf("Input: n = 1\n");

printf("output: ");

nthMagicNo(1);

printf("\nInput: n = 2\n");

printf("output: ");

nthMagicNo(2);

printf("\nInput: n = 3\n");

printf("output: ");

nthMagicNo(3);

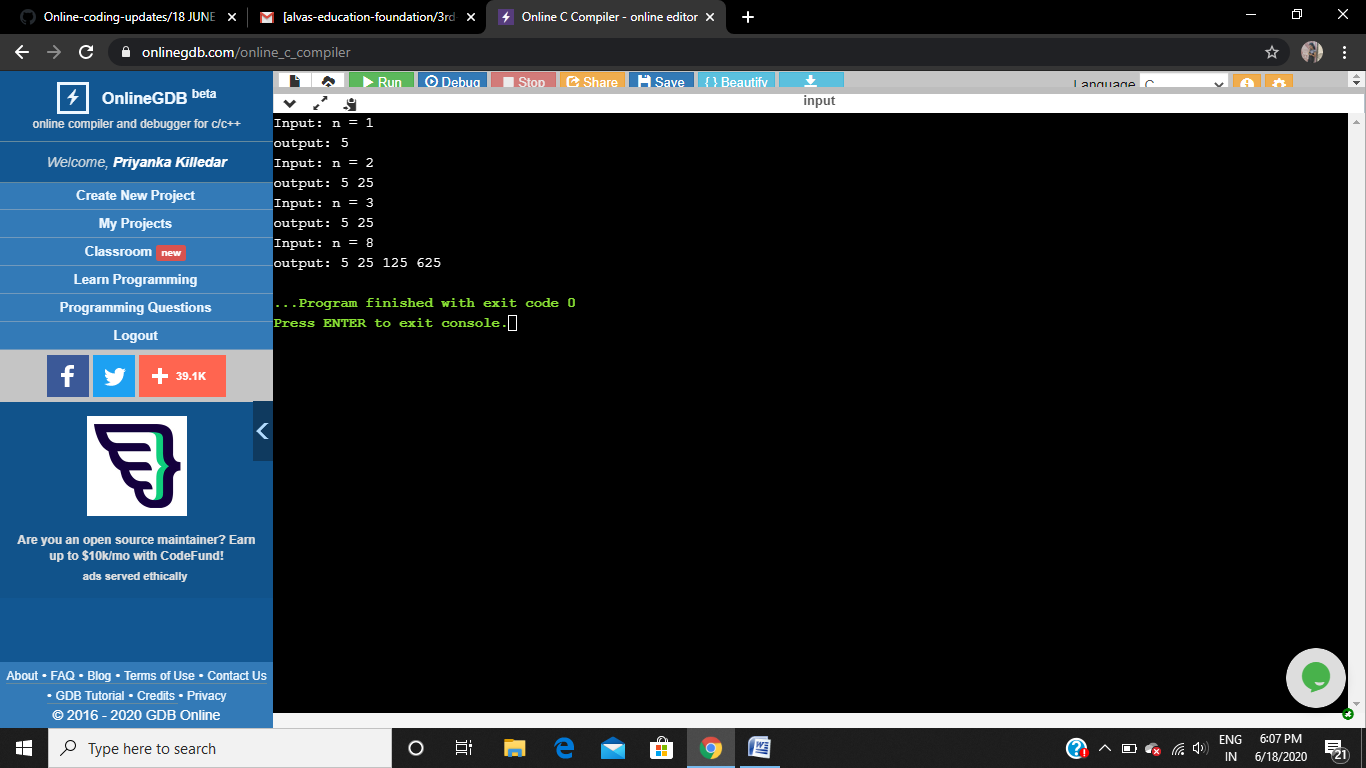
printf("\nInput: n = 8\n");

printf("output: ");

nthMagicNo(8);

return 0;

}



**Write a Java program to Check if a binary tree is binary search tree or not**

class Node

{

int data;

Node left, right;

public Node(int item)

{

data = item;

left = right = null;

}

}

public class BinaryTree

{

Node root;

boolean isBST() {

return isBSTUtil(root, Integer.MIN\_VALUE,

Integer.MAX\_VALUE);

}

boolean isBSTUtil(Node node, int min, int max)

{

if (node == null)

return true;

if (node.data < min || node.data > max)

return false;

return (isBSTUtil(node.left, min, node.data-1) &&

isBSTUtil(node.right, node.data+1, max));

}

public static void main(String args[])

{

BinaryTree tree = new BinaryTree();

tree.root = new Node(7);

tree.root.left = new Node(2);

tree.root.right = new Node(5);

tree.root.left.left = new Node(1);

tree.root.left.right = new Node(3);

if (tree.isBST())

System.out.println("IS BST");

else

System.out.println("Not a BST");

}

}

**Node= 4,2,5,1,3 IS BST Node= 7,2,5,1,3 IS NOT BST**

