**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **18-06-2020** | | | | | **Name:** | **Ashika** | |
| **Sem & Sec** | **6th  Sem ‘A’ Sec** | | | | | **USN:** | **4AL17CS016** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **C programming Quiz and python workshop quiz** | | | | | | |
| **Max. Marks** | | **--** | | **Score** | | | **--** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Workshop exercise solving** | | | | | | | |
| **Certificate Provider** | | | **--** | | **Duration** | | | **--** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement: 2 programs** | | | | | | | | |
| **Status: done** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | <https://github.com/ASHIKA-05/DAILY-REPORT> | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

**PYTHON WORKSHOP:**

Refer the github link for exercise programs: :- <https://github.com/ASHIKA-05/workshop-ML-and-python>

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

import java.util.\*;

class Node

{

int data;

Node left = null, right = null;

Node(int data) {

this.data = data;

}

}

class Main

{

public static Node insert(Node root, int key)

{

if (root == null) {

return new Node(key);

}

if (key < root.data) {

root.left = insert(root.left, key);

}

else {

root.right = insert(root.right, key);

}

return root;

}

public static boolean isBST(Node node, int minKey, int maxKey)

{

if (node == null) {

return true;

}

if (node.data < minKey || node.data > maxKey) {

return false;

}

return isBST(node.left, minKey, node.data) &&

isBST(node.right, node.data, maxKey);

}

public static void isBST(Node root)

{

if (isBST(root, Integer.MIN\_VALUE, Integer.MAX\_VALUE)) {

System.out.println("This is a BST.");

} else {

System.out.println("This is NOT a BST!");

}

}

private static void swap(Node root) {

Node left = root.left;

root.left = root.right;

root.right = left;

}

public static void main(String[] args)

{

Node root = null;

int i;

int[] keys = new int[10];

Scanner s = new Scanner(System.in);

System.out.println("enter the no of elements :");

int n=s.nextInt();

System.out.println("enter the elements :");

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

{

keys[i]=s.nextInt();

}

for (int key : keys) {

root = insert(root, key);

}

swap(root);

isBST(root);

}

}

2) [Write a C Program to generate first N Magic Numbers.](https://github.com/orgs/alvas-education-foundation/teams/3rd-year/discussions/95)

#include <stdio.h>

#define max 1000000007

long long int magic(int n){

long long int pro=1;

long long res=0;

while(n){

pro=(pro\*5)%max;

if(n&1)

res=(res+pro)%max;

n=n>>1;

}

return res;

}

int main()

{

int n,i;

printf("Enter n :");

scanf("%d",&n);

printf(" the magic numbers are: ");

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

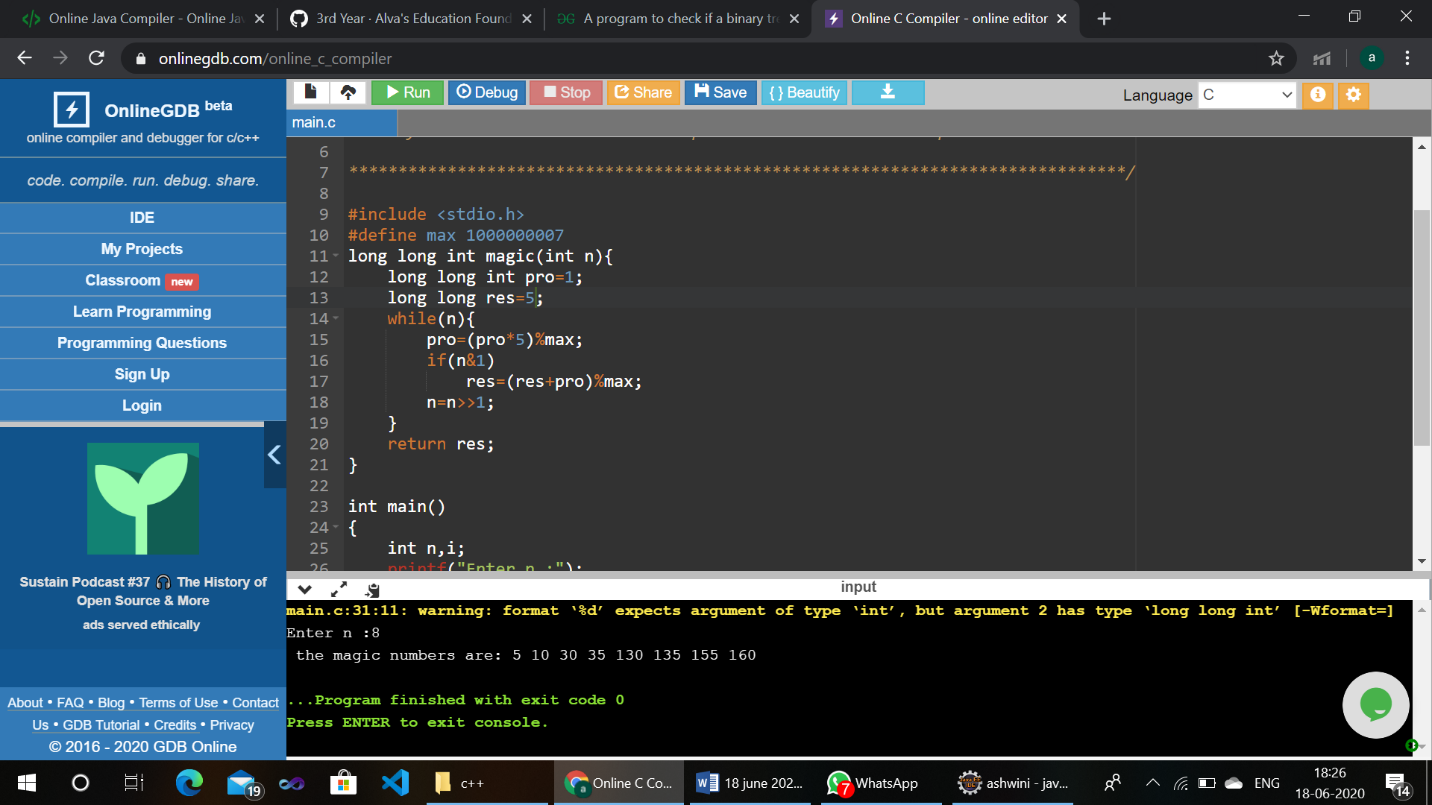
{

printf("%d ",magic(i));

}

return 0;

}

****

