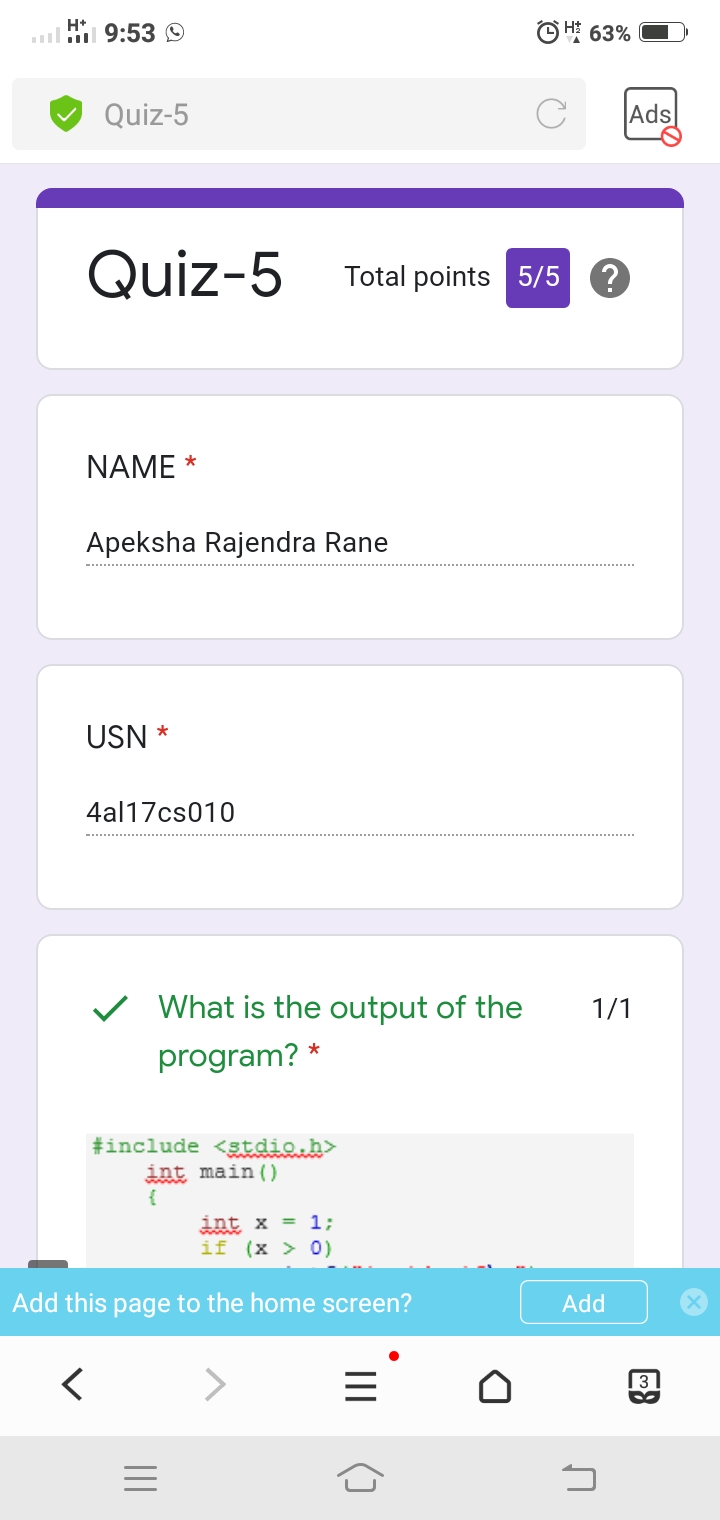
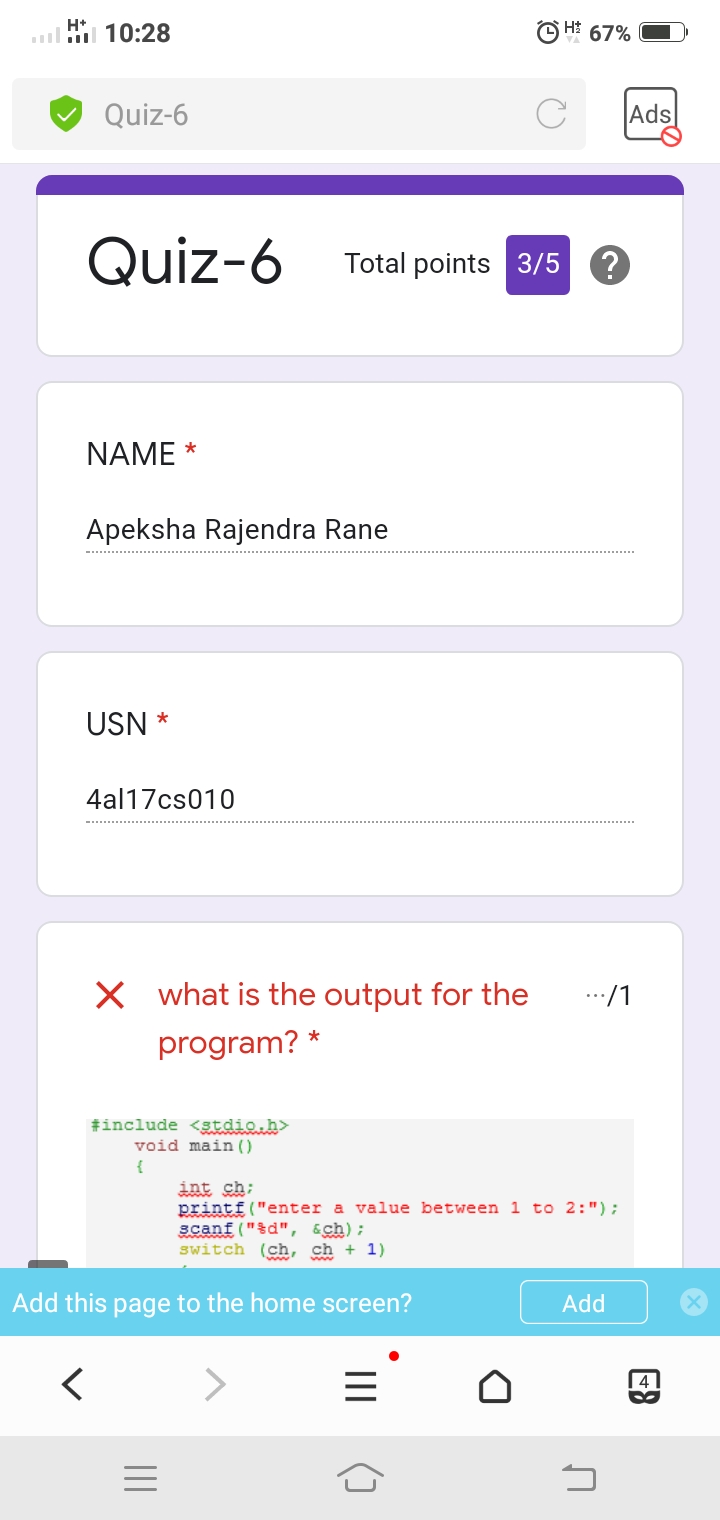
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **18-06-2020** | | | | | **Name:** | | **Apeksha Rane** | |
| **Sem & Sec** | **6th  Sem ‘A’ Sec** | | | | | **USN:** | | **4AL17CS010** | |
| **Online Test Summary** | | | | | | | | | |
| **Subject** | | **C programming Quiz and**  **python workshop quiz** | | | | | | | |
| **Max. Marks** | | **C quiz1=5**  **Cquiz2=5** | | **Score** | | | | **Quiz1=5**  **Quiz2=3** | |
| **Pre-Placement Training Summary** | | | | | | | | | |
| **Course** | **Workshop of C programming and Python workshop.** | | | | | | | | |
| **Faculty** | | | **Vivek Sharma sir.**  **Dr.Badhusha sir.** | | | | **Duration** | | **4 hours** |
| **Coding Challenges** | | | | | | | | | |
| **Problem Statement:1.Python workshop workout examples and Exercises .**  **2. Write a Java program to Check if a binary tree is binary search tree or not.**  **3. Write a C Program to generate first N Magic Numbers.** | | | | | | | | | |
| **Status: done** | | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | | |
| **If yes Repository name** | | | | | Daily Report =<https://github.com/Apeksha12appu/19-5-2020-online-coding-activity>  Python workshop= <https://github.com/Apeksha12appu/python-using-DA-and-ML> | | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | | |

**Class and Quiz Snapshots:**

**C programming:**

****

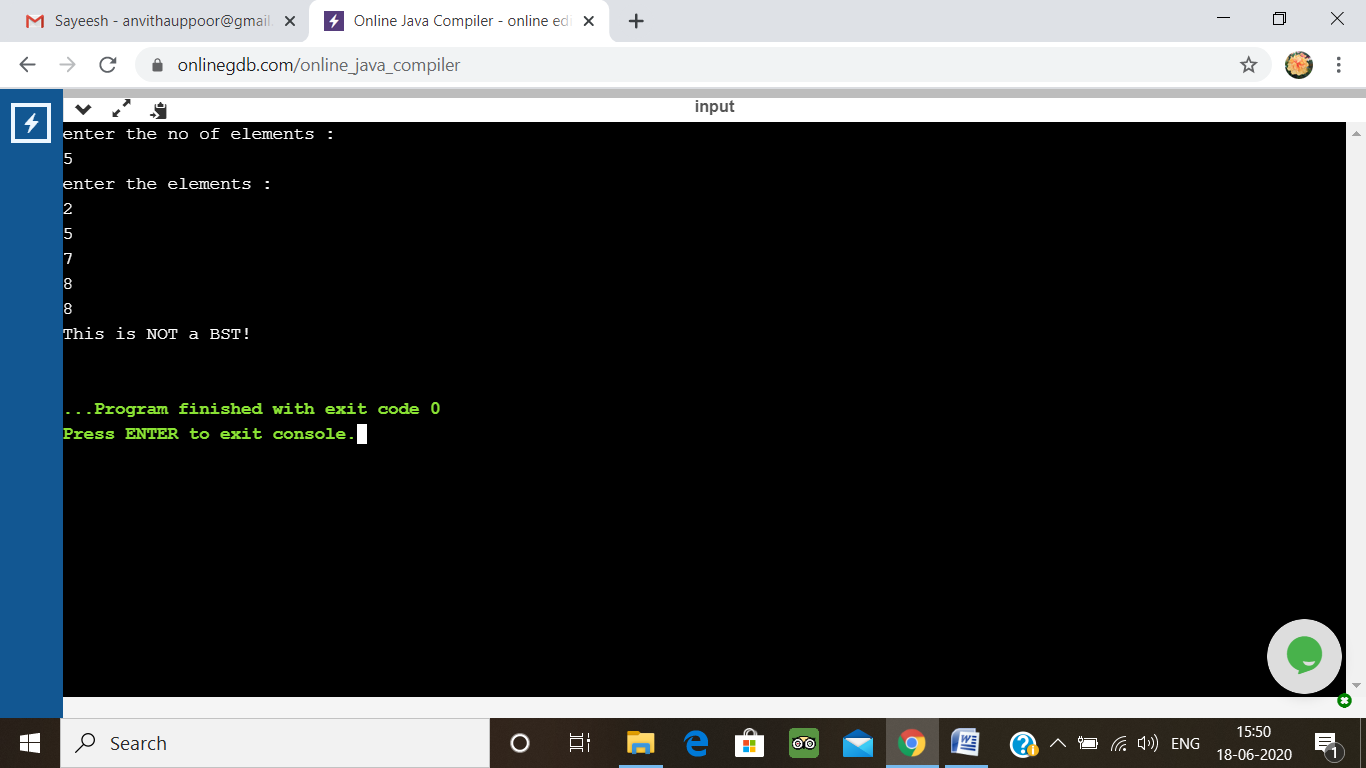
****

**Coding Challenge:**

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);  }  }   |  |  | | --- | --- | |  |  | |  |  |

**Output:**



2. Write a C Program to generate first N Magic Numbers.

A magic number is defined as a number which can be expressed as a power of 5 or sum of unique powers of 5. First few magic numbers are 5, 25, 30(5 + 25), 125, 130(125 + 5), ….  
Input: n = 1  
Output: 5

Input: n = 2  
Output: 5 25

Input: n = 3  
Output: 5 25 30

Input: n = 8  
Output: 5 25 30 125 130 150 155 625

Hint:  
The magic numbers can be represented as 001, 010, 011, 100, 101, 110 etc, where 001 is 0pow(5,3) + 0pow(5,2) + 1\*pow(5,1). So basically, we need to add powers of 5 for each bit set in given integer n.  
If n = 1; binary representation of 1 = 0001  
Magic Number is: 0 \* pow(5,4) + 0 \* pow(5, 3) + 0 \* pow(5, 2) + 1 \* pow(5, 1) = 5

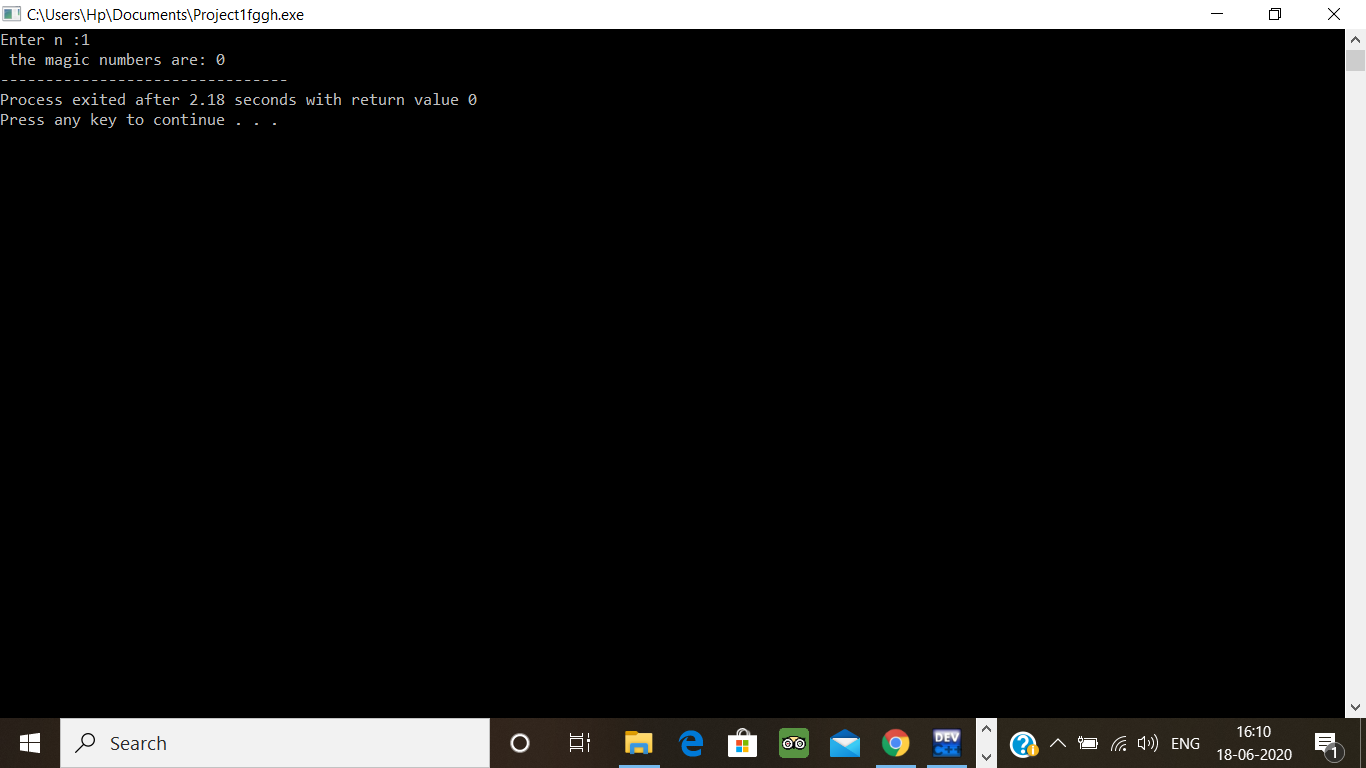
If n = 6; Binary representation of 6 is 0110  
6th Magic Number is: 0 \* pow(5, 4) + 1 \* pow(5, 3) + 1 \* pow(5, 2) + 0 \* pow(5, 1)  
= 0 + 125 + 25 + 0 = 150

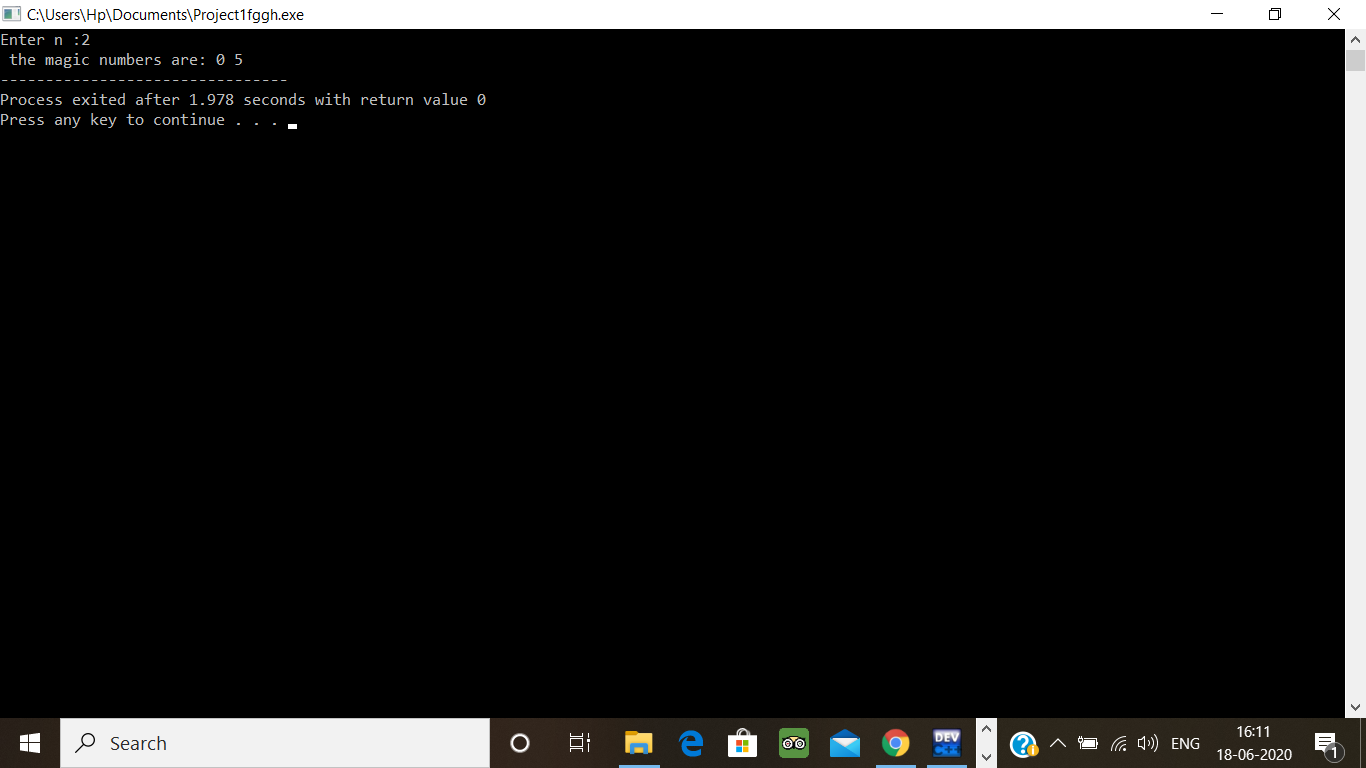
Logic:

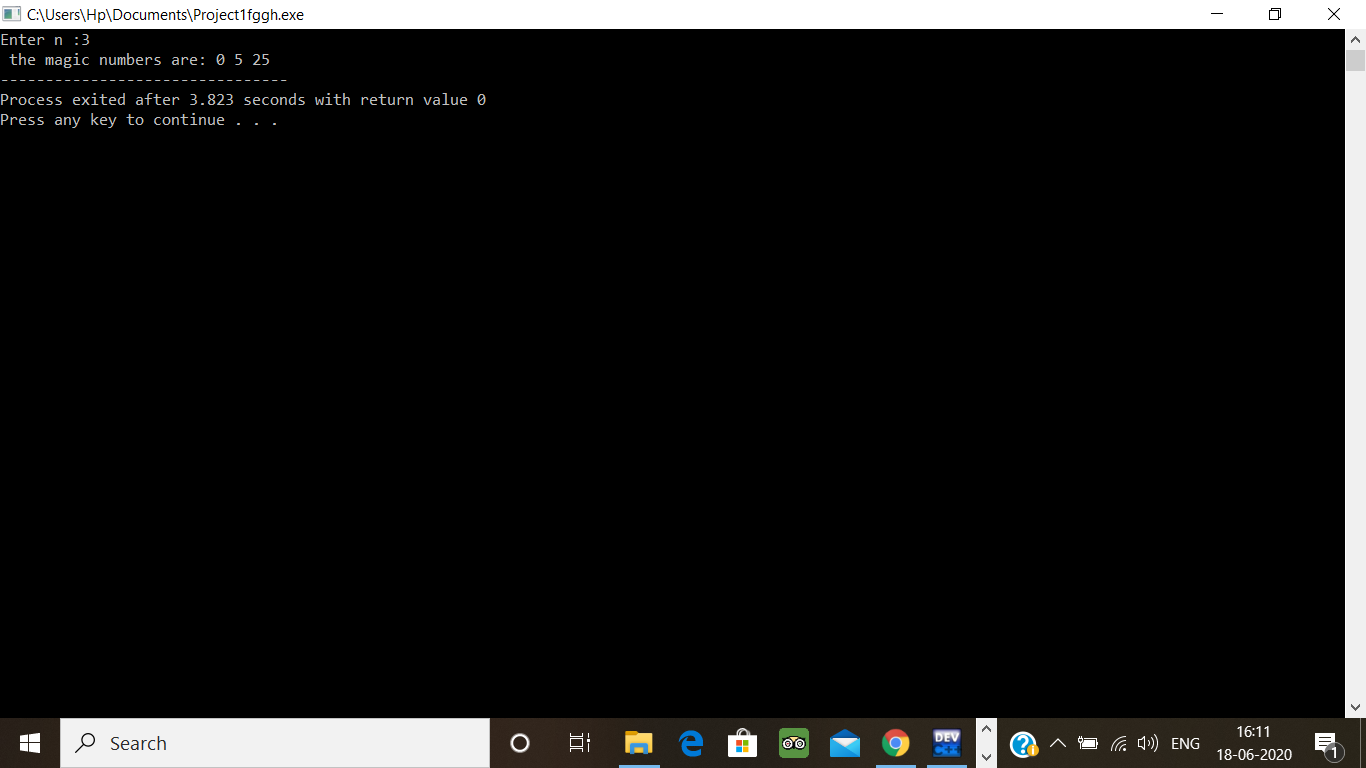
Read n  
for(i = 0 to n)  
{  
Display ith magic number  
}

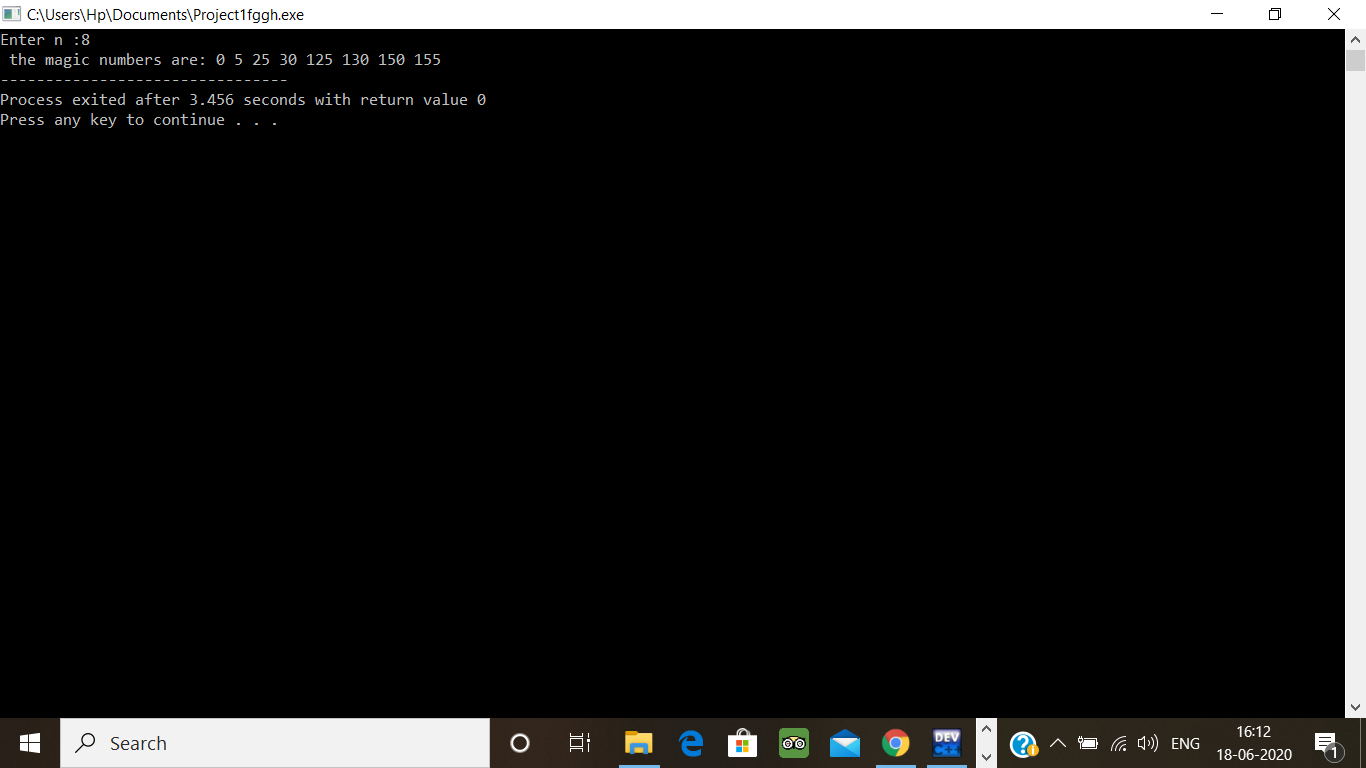
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| #include <stdio.h>  #define max 1000000007  long long int magic(int n){  long long int pro=1;  long long res=0;  while(n){  pro=(pro\*5)%max;  if(n&1)  res=(res+pro)%max;  n=n>>1;  }  return res;  }  int main()  {  int n,i;  printf("Enter n :");  scanf("%d",&n);  printf(" the magic numbers are: ");  for(i=0;i<n;i++)  {  printf("%d ",magic(i));  }  return 0;  }   |  |  | | --- | --- | |  |  | |  |  |

**Output:**

****

****

****

****