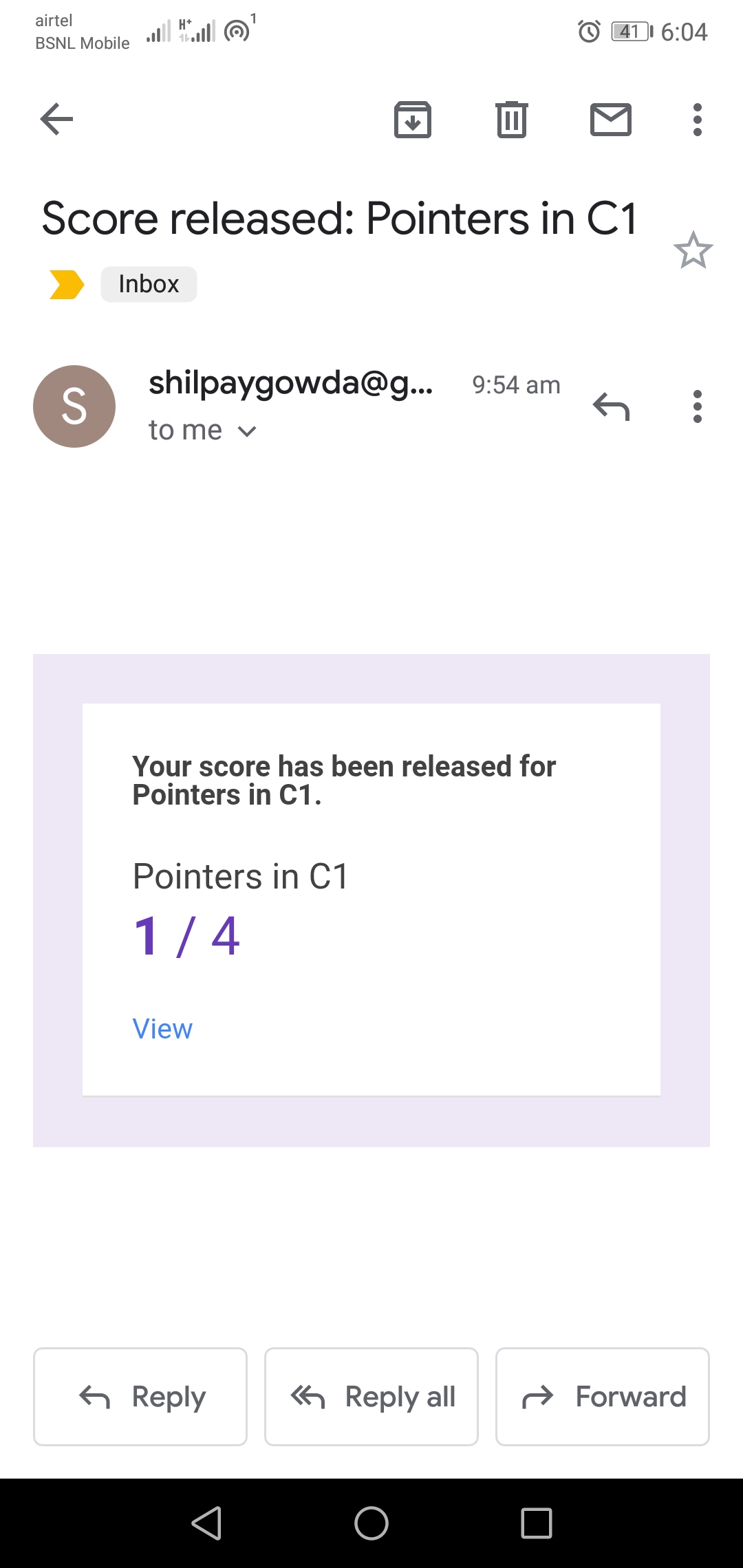
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **19-06-2020** | | | | | **Name:** | | **Anvitha U** | |
| **Sem & Sec** | **6th  Sem ‘A’ Sec** | | | | | **USN:** | | **4AL17CS009** | |
| **Online Test Summary** | | | | | | | | | |
| **Subject** | | **C programming Quiz and**  **python workshop quiz** | | | | | | | |
| **Max. Marks** | | **C quiz1=4**  **Cquiz2=4**  **Cquiz3=4** | | **Score** | | | | **Quiz1=1.**  **Quiz2=not evaluated.**  **Quiz3=not evaluated.** | |
| **Pre-Placement Training Summary** | | | | | | | | | |
| **Course** | **Workshop of C programming and Python workshop.** | | | | | | | | |
| **Faculty** | | | **Shilpa mam.**  **Dr.Badhusha sir.** | | | | **Duration** | | **4 hours** |
| **Coding Challenges** | | | | | | | | | |
| **Problem Statement:1.Python workshop workout examples and Exercises .**  **2.** To swap 2 numbers using pointer and function concept, return value from user defined function and print the swapped values in main() function.  **3.** Given a positive integer n, count the total number of set bits in binary representation of all numbers from 1 to n.  4. Write a Java program to create a doubly linked list of n nodes and display it in reverse order.  . | | | | | | | | | |
| **Status: done** | | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | | |
| **If yes Repository name** | | | | | Daily Report =<https://github.com/anvithauppoor/online_coding_activity>  Python workshop= <https://github.com/anvithauppoor/Python_using_DA_and_ML> | | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | | |

**Class and Quiz Snapshots:**

**C programming:**

****

**Coding Challenge:**

1 **.** To swap 2 numbers using pointer and function concept, return value from user defined function and print the swapped values in main() function.

#include <stdio.h>

void swap(int \*x,int \*y)

{

int t;

t = \*x;

\*x = \*y;

\*y = t;

}

int main()

{

int num1,num2;

printf("Enter value of num1: ");

scanf("%d",&num1);

printf("Enter value of num2: ");

scanf("%d",&num2);

printf("Before Swapping: num1 is: %d, num2 is: %d\n",num1,num2);

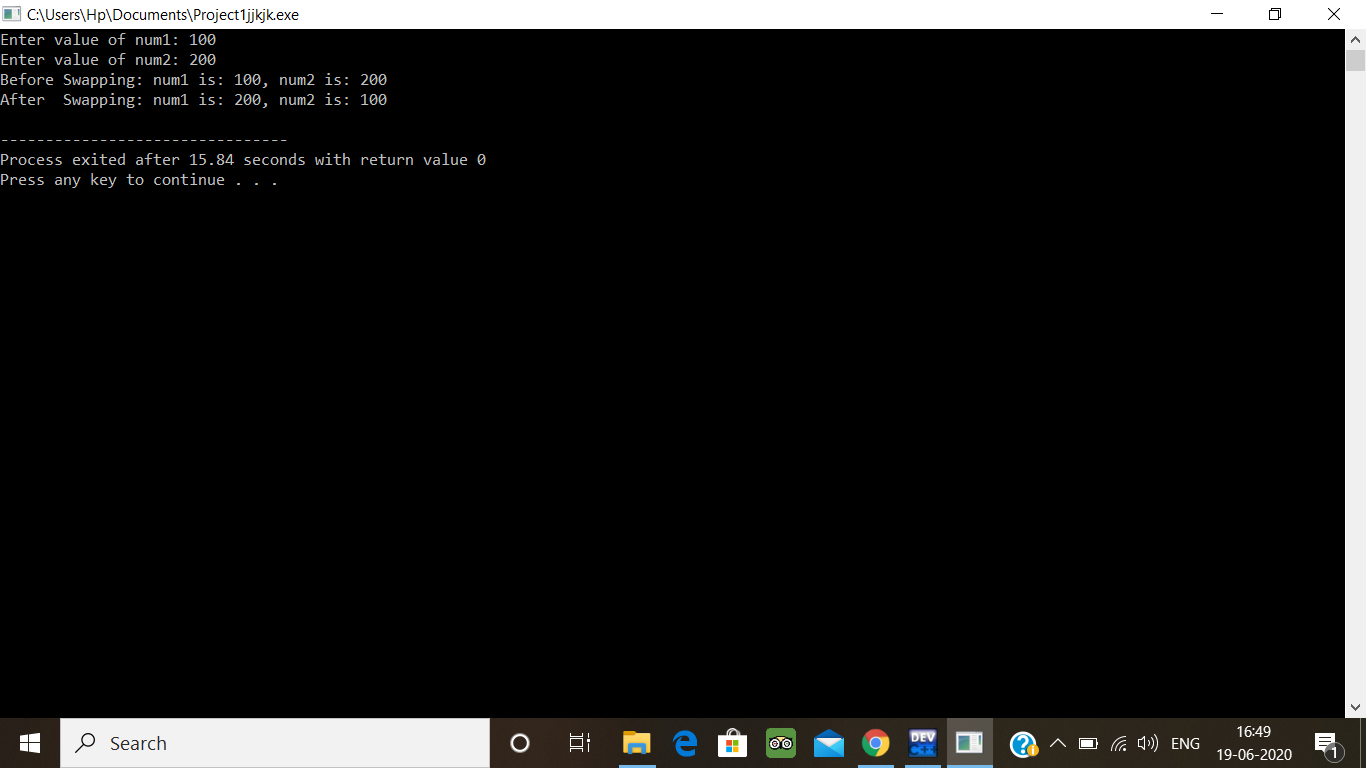
swap(&num1,&num2);

printf("After Swapping: num1 is: %d, num2 is: %d\n",num1,num2);

return 0;

}

**Output:**



**2.** Given a positive integer n, count the total number of set bits in binary representation of all numbers from 1 to n.

#include <stdio.h>

unsigned int getLeftmostBit(int n)

{

int m = 0;

while (n > 1) {

n = n >> 1;

m++;

}

return m;

}

unsigned int getNextLeftmostBit(int n, int m)

{

unsigned int temp = 1 << m;

while (n < temp) {

temp = temp >> 1;

m--;

}

return m;

} unsigned int \_countSetBits(unsigned int n, int m);

unsigned int countSetBits(unsigned int n)

{

int m = getLeftmostBit(n);

return \_countSetBits(n, m);

}

unsigned int \_countSetBits(unsigned int n, int m)

{

if (n == 0)

return 0; m = getNextLeftmostBit(n, m);

if (n == ((unsigned int)1 << (m + 1)) - 1)

return (unsigned int)(m + 1) \* (1 << m);

n = n - (1 << m);

return (n + 1) + countSetBits(n) + m \* (1 << (m - 1));

}

int main()

{

int n = 3;

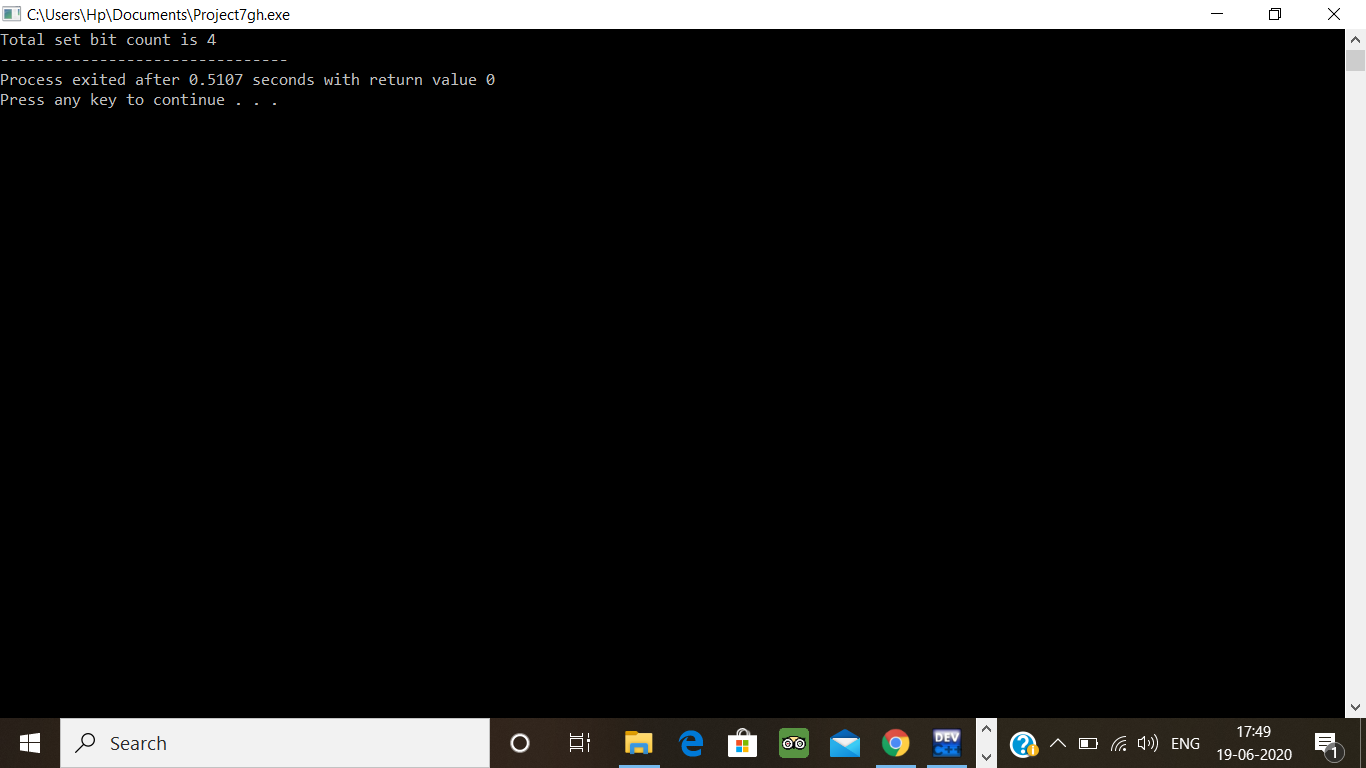
printf("Total set bit count is %d", countSetBits(n));

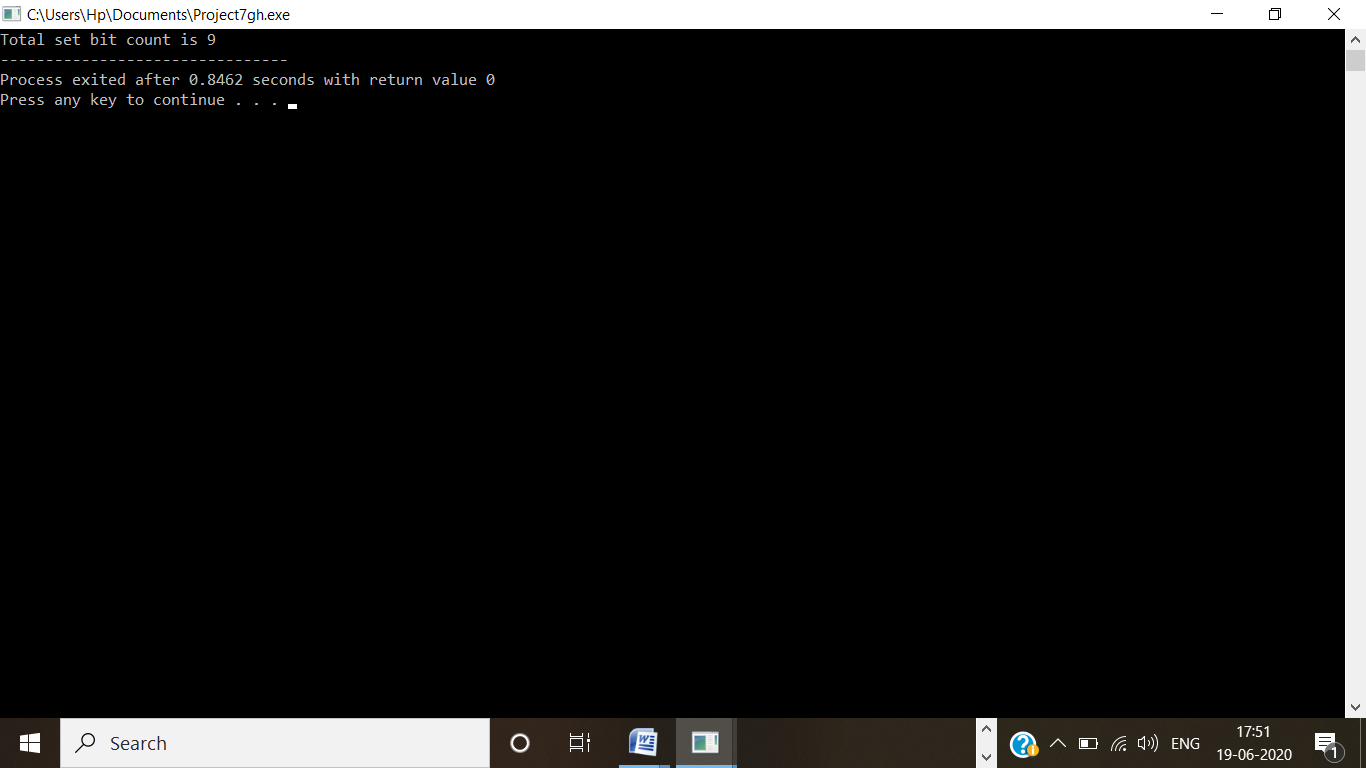
return 0;

}

**Examples:**  
Input: n = 3  
Output: 4  
Input: n = 6  
Output: 9

**Hint:** Read a positive integer (example: 3 indicates range), so u have to consider 1, 2, 3 as the input convert these numbers into binary and count the number of 1 in that (1- 0001, 2- 0010, 3- 0011) number of 1s from all 3 digit is 4 so the answer is 4





3. Write a Java program to create a doubly linked list of n nodes and display it in reverse order

public class ReverseList {

class Node{

int data;

Node previous;

Node next;

public Node(int data) {

this.data = data;

}

}

Node head, tail = null;

public void addNode(int data) {

Node newNode = new Node(data);

if(head == null) {

head = tail = newNode;

head.previous = null;

tail.next = null;

}

else {

tail.next = newNode;

newNode.previous = tail;

tail = newNode;

tail.next = null;

}

}

public void reverse() {

Node current = head, temp = null;

while(current != null) {

temp = current.next;

current.next = current.previous;

current.previous = temp;

current = current.previous;

}

temp = head;

head = tail;

tail = temp;

}

public void display() {

Node current = head;

if(head == null) {

System.out.println("List is empty");

return;

}

while(current != null) {

System.out.print(current.data + " ");

current = current.next;

}

}

public static void main(String[] args) {

ReverseList dList = new ReverseList();

//Add nodes to the list

dList.addNode(1);

dList.addNode(2);

dList.addNode(3);

dList.addNode(4);

dList.addNode(5);

System.out.println("Original List: ");

dList.display();

//Reverse the given list

dList.reverse();

//Displays the reversed list

System.out.println("\nReversed List: ");

dList.display();

}

}

**Output:**

