DAILY ONLINE ACTIVITIES SUMMARY

Date:	**11-07-2020**	**Name:**	**Manikya K**
Sem & Sec	**8th,A**	**USN:**	**4AL16CS050**

Online Test Summary

Subject	**Not Conducted**		
Max. Marks		**Score**	

Certification Course Summary

Course	**1) Robotic Process Automation (RPA)** **2) Introduction to ethical hacking** **3) Introduction to cyber security** **4) Introduction to Hadoop**		
Certificate Provider	**1) GUVI** **2) Great learner academy**	**Duration**	**RPA – 4 Hrs** **Ethical hacking - 6 Hrs** **Cyber Security - 7 Hrs** **Hadoop – 4 Hrs**

Coding Challenges

Problem Statement: `Python Program implementation of binary insertion sort`

Status: Solved

Uploaded the report in Github	**Yes**
If yes Repository name	**manikya-20**
Uploaded the report in slack	**Yes**

Online Test Details: (Attach the snapshot and briefly write the report for the same)

Certification Course Details: (Attach the snapshot and briefly write the report for the same)

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

1) Certification Course Details:

A) Robotic process Automation:

B) Introdution to ethical hacking:

C) Introduction to Cyber Security:

D) Introduction to Hadoop:

2) Coding Challenges:

```
def binary_search(arr, val, start, end):
    # we need to distinugish whether we should insert
    # before or after the left boundary.
    # imagine [0] is the last step of the binary search
    # and we need to decide where to insert -1
    if start == end:
        if arr[start] > val:
            return start
        else:
            return start+1


    # this occurs if we are moving beyond left\'s boundary
    # meaning the left boundary is the least position to
    # find a number greater than val
    if start > end:
        return start


    mid = (start+end)/2
    if arr[mid] < val:
        return binary_search(arr, val, mid+1, end)
    elif arr[mid] > val:
        return binary_search(arr, val, start, mid-1)
    else:
        return mid


def insertion_sort(arr):
    for i in xrange(1, len(arr)):
        val = arr[i]
        j = binary_search(arr, val, 0, i-1)
        arr = arr[:j] + [val] + arr[j:i] + arr[i+1:]
    return arr


print("Sorted array:")
print insertion_sort([37, 23, 0, 17, 12, 72, 31,
                      46, 100, 88, 54])
```