DAILY ONLINE ACTIVITIES SUMMARY

Date:	20-05-2020	Name:	Ainab
Sem & Sec	VIII Semester & A Section	USN:	4AL16CS004

Online Test Summary

Subject	Internet Of Things		
Max. Marks	30	Score	26

Certification Course Summary

Course	Introduction to Hadoop		
Certificate Provider	Great Learning	Duration	14mins

Coding Challenges

Problem Statement: Reverse of the linked list

Status: COMPLETED

Uploaded the report in Github	YES
If yes Repository name	Ainab-16cs004
Uploaded the report in slack	YES

Online Test Details:

, your MCQ result is ready Inbox ☆

TechGig 9:30 AM
to me ⌄

TECHGIG

Hi ,

You have scored **26 marks** in MCQ.

See Assessment

About The Assessment

IOT IA1
Round 1 ends on: 20 May, 2020

Warm Regards,
TechGig Team

2020 | TechGig | Terms of Use | Contact Us
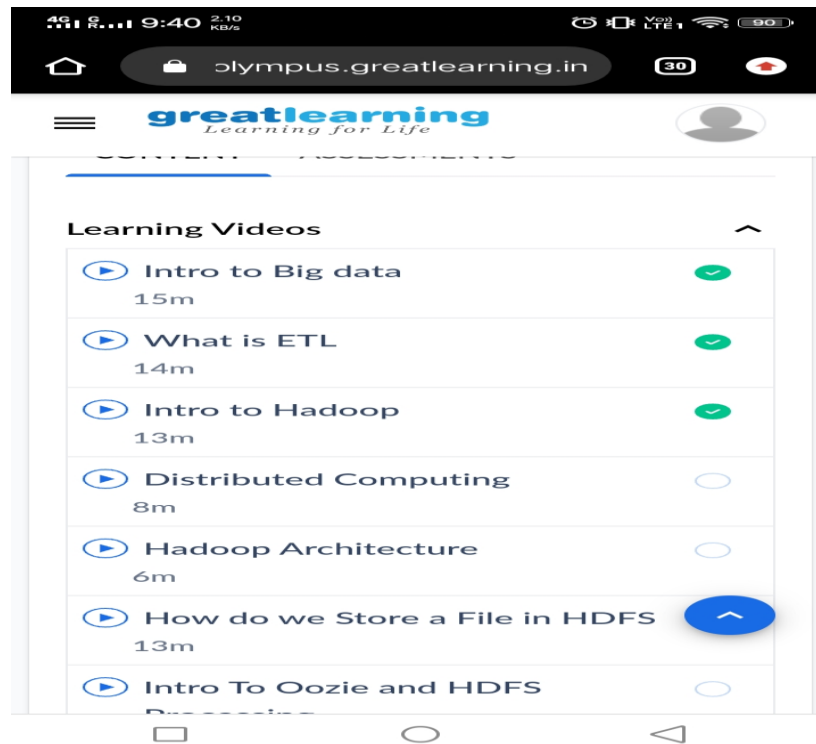Times Center, FC – 6, Sector 16 A, Film City,
Noida – 201301, Uttar Pradesh, India
Follow Us on   Download App

Certification Course Details:

Intro to Hadoop

The Hadoop take a bunch of machines, while installing Hadoop will ask which is master and which is slave. Hadoop is a framework in which we can expand to any number of machine and due to Hadoop the user does not have to worry about the storage because it can resize without downtime. We can add machine and also remove it only if there is no program running in the machine. This whole process is called as Hadoop Clustering. Clustering means group of machines. The Hadoop clusters can also be built in desktops.

Coding Challenges Details:

If a linked listis: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8
The value of size k is 2
Then the linked list looks like: 2 → 1 → 4 → 3 → 6 → 5 → 8 → 7

If a linked listis: 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8
The value of size k is 3
Then the linked list looks like: 3 → 2 → 1 → 6 → 5 → 4 → 8 → 7

Solution:

```c
struct Node
{
int data;
struct Node* next;
};
pointer to the new head node. /
struct Node reverse (struct Node head, int k)
{
struct Node current = head;
struct Node next = NULL;
struct Node prev = NULL;
int count = 0;
while (current != NULL && count < k) { next = current->next; current-
>next = prev; prev = current; current = next; count++; } if (next != NULL)
head->next = reverse(next, k); return prev;
}
void push(struct Node** head_ref, int new_data)
{
struct Node* new_node =
(struct Node*) malloc(sizeof(struct Node));
new_node->data = new_data; new_node->next = (*head_ref); (*head_ref)
= new_node;
}
void printList(struct Node *node)
{
while (node != NULL)
{
printf("%d ", node->data);
```

```c
            node = node->next;
        }
    }
int main(void)
{
struct Node* head = NULL;
push(&head, 8);
push(&head, 7);
push(&head, 6);
push(&head, 5);
push(&head, 4);
push(&head, 3);
push(&head, 2);
push(&head, 1);
printf("\nGiven linked list \n"); printList(head); head = reverse(head, 2);
printf("\nReversed Linked list \n"); printList(head); return(0);
}
```