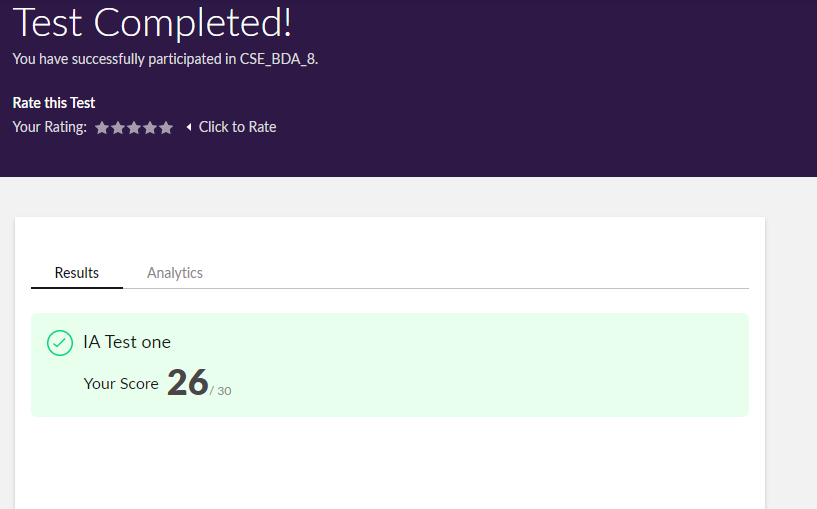
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **16-06-2020** | | | | | **Name:** | **Deeksha D Poojary** | |
| **Sem & Sec** | **VIII Semester & A Section** | | | | | **USN:** | **4AL16CS026** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **BDA** | | | | | | |
| **Max. Marks** | | **30** | | **Score** | | | **26** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Introduction to Amazon Elastic Compute Cloud (EC2)** | | | | | | | |
| **Certificate Provider** | | | **Amazon Web Service** | | **Duration** | | | **10 minutes** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement: Write a program in for triple linked list** | | | | | | | | |
| **Status: COMPLETED** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **YES** | | | |
| **If yes Repository name** | | | | | **deekshapoojari** | | | |
| **Uploaded the report in slack** | | | | | **YES** | | | |

Online Test Details:



Certification Course Coding Challenges Details:

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**Program1:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | |  |
| **#include<iostream>** |
|  | **#include<stdio.h>** | |
|  |  | |
|  | **using namespace std;** | |
|  |  | |
|  | **int a = 0;** | |
|  |  | |
|  | **struct node** | |
|  | **{** | |
|  | **node \*next, \*prev, \*top;** | |
|  | **int info;** | |
|  | **}\*head = NULL, \*tail = NULL, \*p = NULL, \*r = NULL, \*np = NULL, \*q = NULL;** | |
|  |  | |
|  | **void create(int z)** | |
|  | **{** | |
|  | **np = new node;** | |
|  | **np->info = z;** | |
|  | **np->next = NULL;** | |
|  | **np->prev = NULL;** | |
|  | **np->top = NULL;** | |
|  | **if (a == 0)** | |
|  | **{** | |
|  | **tail = np;** | |
|  | **head = np;** | |
|  | **p = head;** | |
|  | **p->next = NULL;** | |
|  | **p->prev = NULL;** | |
|  | **p->top = NULL;** | |
|  | **a++;** | |
|  | **}** | |
|  | **else** | |
|  | **{** | |
|  | **p = head;** | |
|  | **r = p;** | |
|  | **if (np->info < p->info)** | |
|  | **{** | |
|  | **np->next = p;** | |
|  | **p->prev = np;** | |
|  | **np->prev = NULL;** | |
|  | **head = np;** | |
|  | **p = head;** | |
|  | **do** | |
|  | **{** | |
|  | **p = p->next;** | |
|  | **}** | |
|  | **while (p->next != NULL);** | |
|  | **tail = p;** | |
|  | **}** | |
|  | **else if (np->info > p->info)** | |
|  | **{** | |
|  | **while (p != NULL && np->info > p->info)** | |
|  | **{** | |
|  | **r = p;** | |
|  | **p = p->next;** | |
|  | **if (p == NULL)** | |
|  | **{** | |
|  | **r->next = np;** | |
|  | **np->prev = r;** | |
|  | **np->next = NULL;** | |
|  | **tail = np;** | |
|  | **break;** | |
|  | **}** | |
|  | **else if (np->info <= p->info)** | |
|  | **{** | |
|  | **if (np->info < p->info)** | |
|  | **{** | |
|  | **r->next = np;** | |
|  | **np->prev = r;** | |
|  | **np->next = p;** | |
|  | **p->prev = np;** | |
|  | **if (p->next != NULL)** | |
|  | **{** | |
|  | **do** | |
|  | **{** | |
|  | **p = p->next;** | |
|  | **}** | |
|  | **while (p->next !=NULL);** | |
|  | **}** | |
|  | **tail = p;** | |
|  | **break;** | |
|  | **}** | |
|  | **else if (p->info == np->info)** | |
|  | **{** | |
|  | **q = p;** | |
|  | **while (q->top != NULL)** | |
|  | **{** | |
|  | **q = q->top;** | |
|  | **}** | |
|  | **q->top = np;** | |
|  | **np->top = NULL;** | |
|  | **break;** | |
|  | **}** | |
|  | **}** | |
|  | **}** | |
|  | **}** | |
|  | **}** | |
|  | **}** | |
|  |  | |
|  | **void traverse\_tail()** | |
|  | **{** | |
|  | **node \*t = tail;** | |
|  |  | |
|  | **while (t != NULL)** | |
|  | **{** | |
|  | **cout<<t->info<<"\t";** | |
|  | **q = t;** | |
|  | **while (q->top != NULL)** | |
|  | **{** | |
|  | **q = q->top;** | |
|  | **cout<<"top->"<<q->info<<"\t";** | |
|  | **}** | |
|  | **t = t->prev;** | |
|  | **}** | |
|  | **cout<<endl<<endl;** | |
|  | **}** | |
|  |  | |
|  | **void traverse\_head()** | |
|  | **{** | |
|  | **node \*t = head;** | |
|  | **while (t != NULL)** | |
|  | **{** | |
|  | **cout<<t->info<<"\t";** | |
|  | **q = t;** | |
|  | **while (q->top != NULL)** | |
|  | **{** | |
|  | **q = q->top;** | |
|  | **cout<<"top->"<<q->info<<"\t";** | |
|  | **}** | |
|  | **t = t->next;** | |
|  | **}** | |
|  | **cout<<endl<<endl;** | |
|  | **}** | |
|  |  | |
|  | **int main()** | |
|  | **{** | |
|  | **int c = 0, no, value, ch;** | |
|  | **cout<<"Please enter the number of nodes: "<<endl;** | |
|  | **cin>>no;** | |
|  | **while (c < no)** | |
|  | **{** | |
|  | **cout<<endl<<"Enter the value of node: "<<endl;** | |
|  | **cin>>value;** | |
|  | **create(value);** | |
|  | **c++;** | |
|  | **}** | |
|  | **cout<<endl<<"Traversing Doubly Linked List head: "<<endl;** | |
|  | **traverse\_head();** | |
|  |  | |
|  | **cout<<endl<<"Traversing Doubly Linked List tail: "<<endl;** | |
|  | **traverse\_tail();** | |
|  |  | |
|  | **}** | |