**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | | **25/06/2020** | | | | **Name:** | **Soundarya R** | |
| **Sem & Sec** | | **6th & B** | | | | **USN:** | **4al17cs096** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | |  | | | | | |
| **Max. Marks** | | |  | **Score** | | |  | |
| **Pre-placement Training Summary** | | | | | | | | |
| **Topic** |  | | | | | | | |
| **Faculty** |  | | | | **Duration** | | |  |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:** 2 programs | | | | | | | | |
| **Status: Solved** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | <https://github.com/ashu102/Daily-Status> | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

ONLINE CODING

**1. C program to implement simple queue using SLL.**

#include <stdio.h>

#include <stdlib.h>

struct QNode

{

int key;

struct QNode\* next;

};

struct Queue

{

struct QNode \*front, \*rear;

};

struct QNode\* newNode(int k)

{

struct QNode\* temp = (struct QNode\*)malloc(sizeof(struct QNode));

temp->key = k;

temp->next = NULL;

return temp;

}

struct Queue\* createQueue()

{

struct Queue\* q = (struct Queue\*)malloc(sizeof(struct Queue));

q->front = q->rear = NULL;

return q;

}

void enQueue(struct Queue\* q, int k)

{

struct QNode\* temp = newNode(k);

if (q->rear == NULL) {

q->front = q->rear = temp;

return;

}

q->rear->next = temp;

q->rear = temp;

}

void deQueue(struct Queue\* q)

{

if (q->front == NULL)

return;

struct QNode\* temp = q->front;

q->front = q->front->next;

if (q->front == NULL)

q->rear = NULL;

free(temp);

}

int main()

{

struct Queue\* q = createQueue();

enQueue(q, 10);

enQueue(q, 20);

deQueue(q);

deQueue(q);

enQueue(q, 30);

enQueue(q, 40);

enQueue(q, 50);

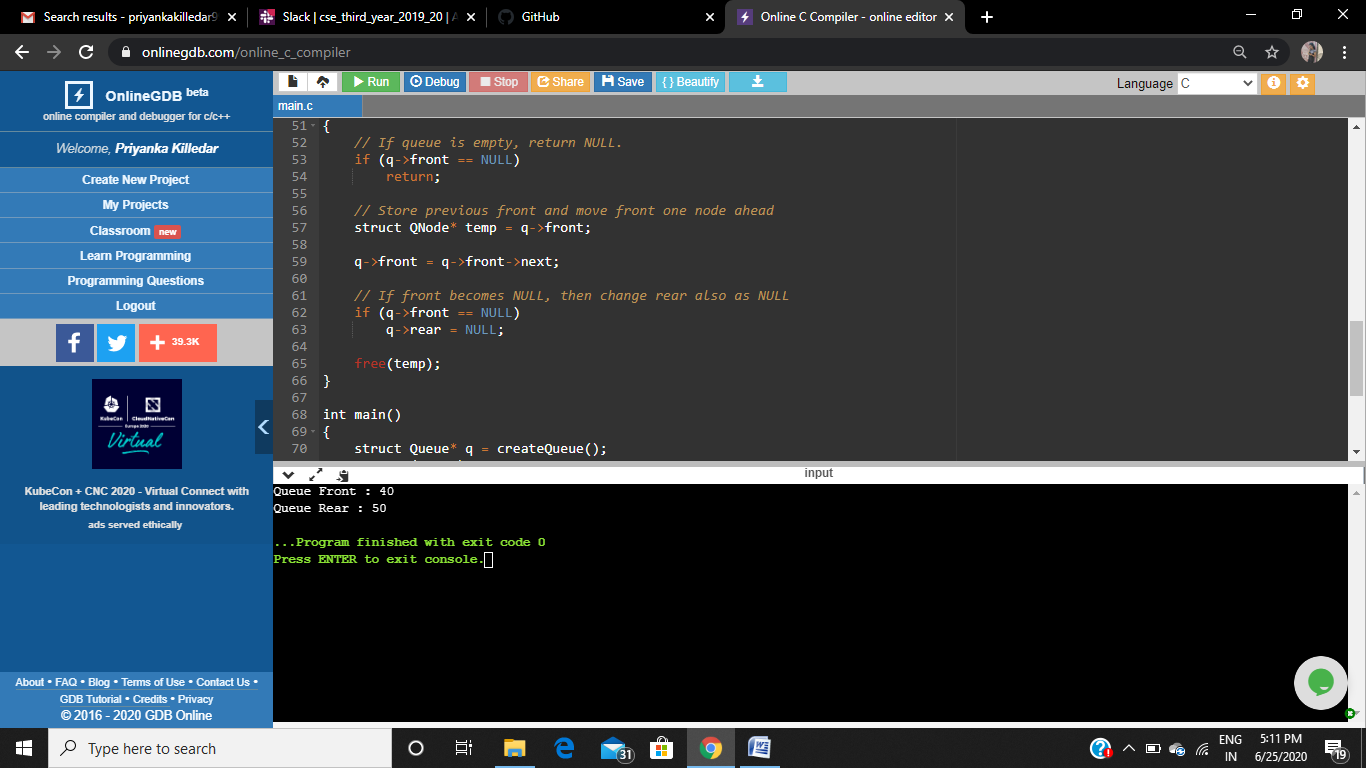
deQueue(q);

printf("Queue Front : %d \n", q->front->key);

printf("Queue Rear : %d", q->rear->key);

return 0;

}



**2. program to Implement a stack using singly linked list**

#include <bits/stdc++.h>

using namespace std;

struct Node

{

int data;

struct Node\* link;

};

struct Node\* top;

void push(int data)

{

struct Node\* temp;

temp = new Node();

if (!temp) {

cout<< "\nHeap Overflow";

exit(1);

}

temp->data = data;

temp->link = top;

top = temp;

}

intisEmpty()

{

return top == NULL;

}

int peek()

{

// check for empty stack

if (!isEmpty())

return top->data;

else

exit(1);

}

void pop()

{

struct Node\* temp;

if (top == NULL) {

cout<< "\nStack Underflow" <<endl;

exit(1);

}

else {

temp = top;

top = top->link;

temp->link = NULL;

free(temp);

}

}

void display()

{

struct Node\* temp;

if (top == NULL) {

cout<< "\nStack Underflow";

exit(1);

}

else {

temp = top;

while (temp != NULL) {

// print node data

cout<< temp->data << " ";

temp = temp->link;

}

}

}

int main()

{

push(11);

push(22);

push(33);

push(44);

display();

cout<< "\nTop element is %d\n" << peek();

pop();

pop();

display();

cout<< "\nTop element is %d\n" << peek();

return 0;

}

