

DAILY ONLINE ACTIVITIES SUMMARY

Date:	22/5/2020	Name:	Divya C H
Sem & Sec	8 th Sem	USN:	4AL16CS033
Online Test Summary			
Subject	Big Data Analytics		
Max. Marks	40	Score	28
Certification Course Summary			
Course	Introduction to Ethical Hacking		
Certificate Provider	greatlearning.in	Duration	6 hrs
Coding Challenges			
Problem Statement: 1)Write a C Program to implement various operations of Singly Linked List Stack			
Status: Completed			
Uploaded the report in Github		Yes	
If yes Repository name		Daily_report	
Uploaded the report in slack		yes	

Online Test Details:

The screenshot shows a Gmail interface with a browser window open to mail.google.com. The email subject is "Divya C H, your Module 2 result is ready". The sender is TechGig <user@techgig.com>. The email content includes the TechGig logo, a greeting "Hi Divya C H," and a message "You have scored 28 marks in Module 2." A red button labeled "See Assessment" is present. Below this, there is a section "About The Assessment" with a placeholder image and the text "CSE_BDA_2" and "Round 1 ends on: 22 May, 2020". The email ends with "Warm Regards, TechGig Team". The left sidebar shows the inbox with 1,777 emails and a chat window with contacts "divya" and "Gagana". The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 19:27 on 22-05-2020.

Certification Course Details:

The screenshot displays the Great Learning website interface. The header includes the "greatlearning" logo, navigation links for "Home" and "Live Sessions", and a "My Courses" button. The main content area features a course titled "Introduction to Ethical Hacking" with a "Course In Progress" status. Below the course title, there are two tabs: "CONTENT" and "ASSESSMENTS". Under the "CONTENT" tab, a section titled "Learning Videos" lists six videos with their durations and completion status (indicated by green checkmarks or empty circles):

Video Title	Duration	Status
Career and Growth Ladder in Ethical Hacking	18m	Completed
Domains and Process Implementation under Ethical Hacking	54m	Completed
Ethical Hacking in Network Architecture-Demonstration	48m	Completed
Ethical Hacking in Web Applications-Demonstration	50m	Completed
Ethical Hacking on Mobile Platforms-Demonstration	34m	Completed
What is Ethical Hacking	50m	Not Completed

At the bottom of the page, there is a "Quiz" section.

Coding Challenges Details:

Program 1:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct node
```

```
{
```

```
    int info;
```

```
    struct node *ptr;
```

```
}*top,*top1,*temp;
```

```
int topelement();
```

```
void push(int data);
```

```
void pop();
```

```
void empty();
```

```
void display();
```

```
void destroy();
```

```
void stack_count();
```

```
void create();
```

```
int count = 0;
```

```
void main()
```

```
{
```

```
    int no, ch, e;
```

```
    while (1)
```

```
    {
```

```

printf("\n 1 - Push\t\t2 - Pop");

printf("\n 3 - Top\t\t4 - Check if Stack Empty");

printf("\n 5 - Exit\t\t6 - Display");

printf("\n 7 - Stack Count\t8 - Destroy stack");

                                printf("\n-----\n");

create();

printf("\nEnter choice : ");

scanf("%d", &ch);


switch (ch)
{
case 1:

    printf("Enter data : ");

    scanf("%d", &no);

    push(no);

    break;

case 2:

    pop();

    break;

case 3:

    if (top == NULL)

        printf("No elements in stack");

    else

    {

        e = topelement();

        printf("\n Top element : %d", e);

    }
}

```

```

                                                                    printf("\n-----\n");

        break;

case 4:

    empty();

    break;

case 5:

    exit(0);

case 6:

    display();

    break;

case 7:

    stack_count();

    break;

case 8:

    destroy();

    break;

default :

    printf(" Wrong choice, Please enter correct choice ");

                                                                    printf("\n-----\n");

    break;

}

}

}

void create()

{

    top = NULL;

}

```

```

void stack_count()
{
    printf("\n No. of elements in stack : %d", count);

                                printf("\n-----\n");
}

void push(int data)
{
    if (top == NULL)
    {
        top =(struct node *)malloc(1*sizeof(struct node));
        top->ptr = NULL;
        top->info = data;
    }
    else
    {
        temp =(struct node *)malloc(1*sizeof(struct node));
        temp->ptr = top;
        temp->info = data;
        top = temp;
    }
    count++;

                                printf("\n-----\n");
}

void display()
{
    top1 =top;

```

```

if (top1 == NULL)
{
    printf("Stack is empty");

                                printf("\n ----- \n");

    return;
}

while (top1 != NULL)
{
    printf("%d", top1->info);

    top1 = top1->ptr;
}

                                printf("\n-----\n");
}

void pop()
{
    top1 = top;

    if (top1 == NULL)
    {
        printf("\n Error : Trying to pop from empty stack");

        return;
    }

    else

        top1 = top1->ptr;

    printf("\n Popped value : %d", top->info);

    free(top);

```

```

    top = top1;
    count--;

    printf("\n-----\n");
}

int topelement()
{
    return(top->info);
}

void empty()
{
    if (top == NULL)
        printf("\n Stack is empty");
    else
        printf("\n Stack is not empty with %d elements", count);
        printf("\n-----\n");
}

void destroy()
{
    top1 = top;

    while (top1 != NULL)
    {
        top1 = top->ptr;
        free(top);
        top = top1;
        top1 = top1->ptr;
    }
}

```



```
free(top1);  
top = NULL;  
  
printf("\n All stack elements destroyed");  
count = 0;  
printf("\n ----- \n");  
}
```