

DAILY ONLINE ACTIVITIES SUMMARY

Date:	08-06-2020	Name:	Anvitha Poojary
Sem & Sec	6A	USN:	4AL17CS008
Online Test Summary			
Subject	CNSC		
Max. Marks	60	Score	39
Certification Course Summary			
Course	Machine Learning with Python		
Certificate Provider	COGNITIVE CLASS .ai	Duration	12hr
Coding Challenges			
Problem Statement: 1. Write C++ program to Check whether a number can be represented as difference of two squares 2. C Program to Generate All the Set Partitions of n Numbers Beginning from 1 and so on 3. Java program to delete a node from the middle of the singly linked list 4. Program program to find whether a string is a palindrome or not			
Status: completed			
Uploaded the report in Github		Yes	
If yes Repository name		https://github.com/anvithapo99/Daily-Report	
Uploaded the report in slack		Yes	

Online test details:

Subject: CNSC

The screenshot shows a web browser window with the address bar displaying `techgig.com/challenge/result/mcq/VWRnMEQvUUN4a3BtR1laMThYeTjxZz09`. The page has two tabs: "Anvitha Poojary, CNSC IA 3 - an..." and "Largest Tech Community | Hacka...". The main content area is titled "Results" and "Analytics". It displays three test results:

- Test 3 submitted**
MCQ
Your Score: **21** / 24
- Test 1 submitted**
MCQ
Your Score: **8** / 16
- Test 2 submitted**
MCQ
Your Score: **10** / 20

The Windows taskbar at the bottom shows the Start button, a search bar, and several application icons. The system tray on the right indicates the time is 04:01 PM on 08-06-2020, with language set to ENG.

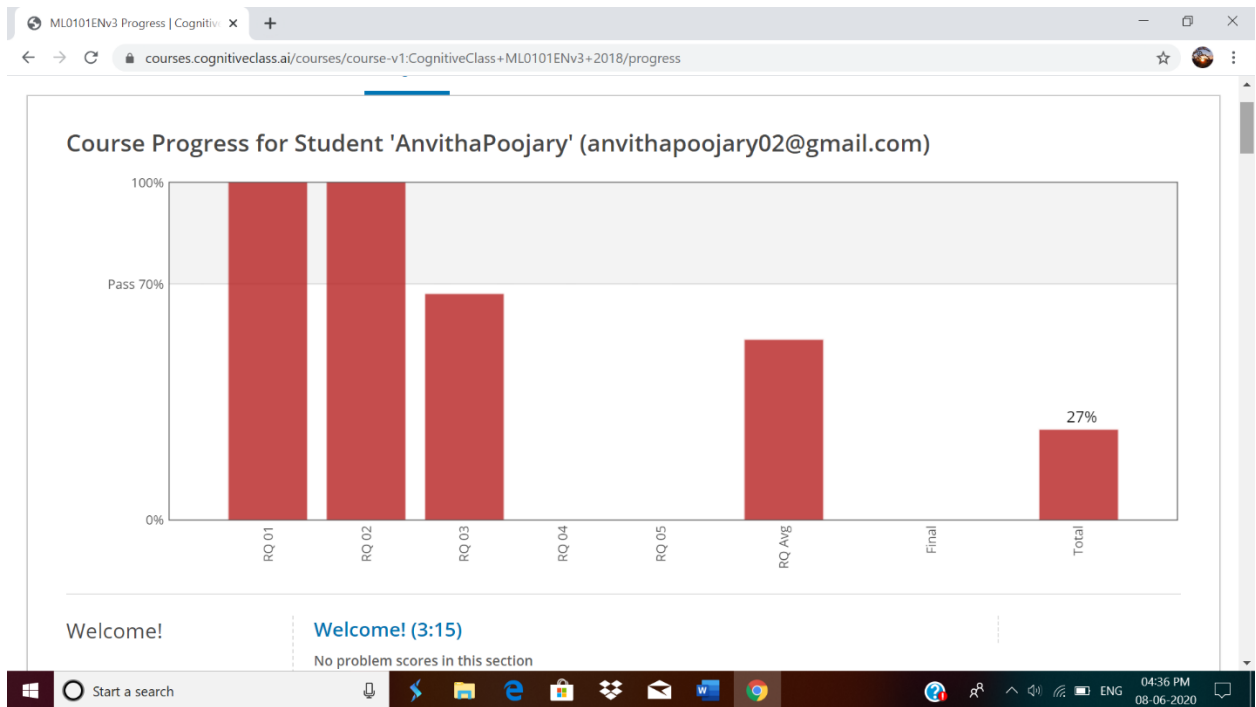
Certification course details:

Machine Learning with Python

Today I have studied following topics:

- Intro to classification
- K nearest neighbor algorithm
- Intro to decision tree

➤ Building decision tree



Coding Challenges Details:

1. Write C++ program to Check whether a number can be represented as difference of two squares

```
#include <iostream>
```

```
using namespace std;
```

```
bool checkPrime(int n);
```

```
int main()
```

```
{
```

```
int n, i;
```

```
bool flag = false;
```

```
cout << "Enter a positive integer: ";
```

```
cin >> n;
```

```
for(i = 2; i <= n/2; ++i)
```

```
{
```

```
    if (checkPrime(i))
```

```
    {
```

```
        if (checkPrime(n - i))
```

```
        {
```

```
            cout << n << " = " << i << " + " << n-i << endl;
```

```
            flag = true;
```

```
        }
```

```
    }
```

```
}
```

```
if (!flag)
```

```
    cout << n << " can't be expressed as sum of two prime numbers.";
```

```
return 0;
```

```
}
```

```
// Check prime number
```

```
bool checkPrime(int n)
{
    int i;
    bool isPrime = true;

    for(i = 2; i <= n/2; ++i)
    {
        if(n % i == 0)
        {
            isPrime = false;
            break;
        }
    }

    return isPrime;
}
```

Output:

The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The code editor displays a C++ program in `main.cpp` with the following content:

```
28
29     return 0;
30 }
31
32 // Check prime number
33 bool checkPrime(int n)
34 {
35     int i;
36     bool isPrime = true;
37     for(i = 2; i <= n/2; ++i)
```

Below the code editor is an input/output window. It shows the prompt "Enter a positive integer: 4", the output "4 = 2 + 2", and a message "...Program finished with exit code 0 Press ENTER to exit console." The Windows taskbar at the bottom shows the date and time as 11:52 AM on 08-06-2020.

2.C Program to Generate All the Set Partitions of n Numbers Beginning from 1 and so on

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

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

```
typedef struct {
```

```
int first;
```

```
int n;
```

```
int level;
```

```
} Call;
```

```
void print(int n, int * a) {
```

```
int i ;
```

```
for (i = 0; i <= n; i++) {
```

```
printf("%d", a[i]);
```

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

```
void integerPartition(int n, int * a){  
  
    int first;  
  
    int i;  
  
    int top = 0;  
  
    int level = 0;  
  
    Call * stack = (Call * ) malloc (sizeof(Call) * 1000);  
  
    stack[0].first = -1;  
  
    stack[0].n = n;  
  
    stack[0].level = level;  
  
    while (top >= 0){  
  
        first = stack[top].first;  
  
        n = stack[top].n;  
  
        level = stack[top].level;  
  
        if (n >= 1) {  
  
            if (first == - 1) {  
  
                a[level] = n;  
  
                print(level, a);  
  
                first = (level == 0) ? 1 : a[level-1];  
  
                i = first;
```

```

    } else {

        i = first;

        i++;

    }

    if (i <= n / 2) {

        a[level] = i;

        stack[top].first = i;

        top++;

        stack[top].first = -1;

        stack[top].n = n - i;

        stack[top].level = level + 1;

    } else {

        top--;

    }

} else {

    top --;

}

}

}

```

```

int main(){

    int N = 1;

    int * a = (int * ) malloc(sizeof(int) * N);

    int i;

```



```

printf("\nEnter a number N to generate all set partition from 1 to N: ");

scanf("%d", &N);

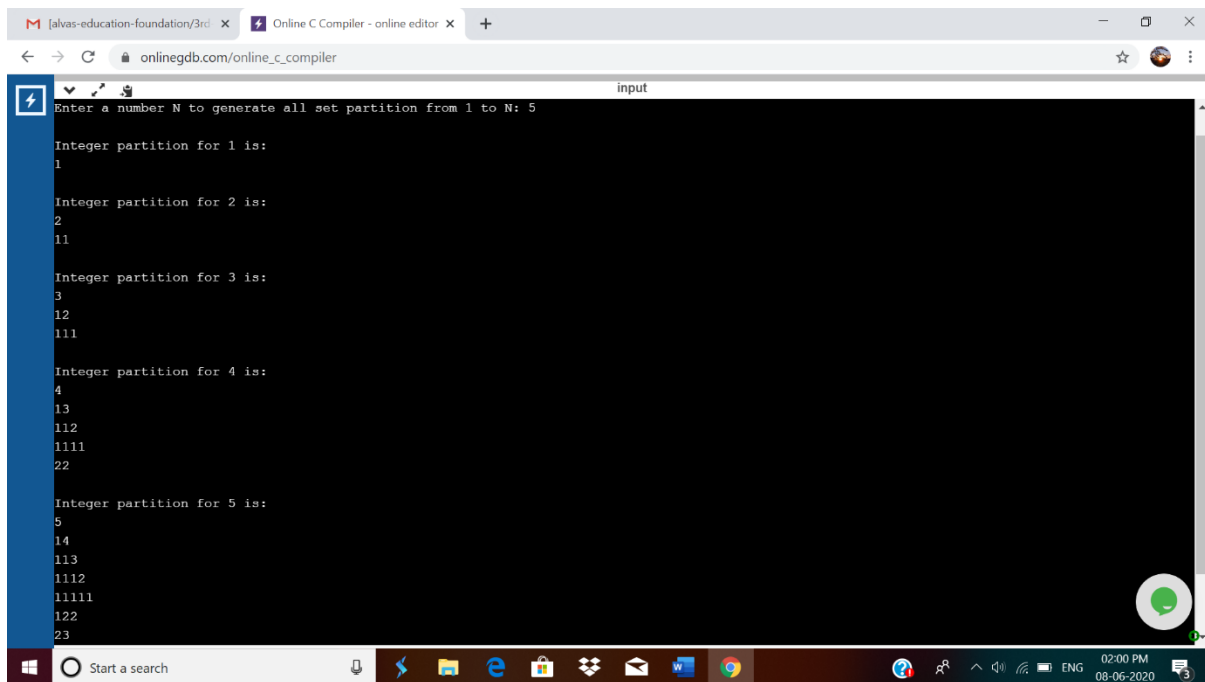
for ( i = 1; i <= N; i++)
{
    printf("\nInteger partition for %d is: \n", i);

    integerPartition (i, a);
}

return(0);
}

```

Output:



The screenshot shows a web browser window with the URL `onlinegdb.com/online_c_compiler`. The browser has two tabs: `[alvas-education-foundation/3rd]` and `Online C Compiler - online editor`. The main content area shows a terminal window with the following output:

```

input
Enter a number N to generate all set partition from 1 to N: 5

Integer partition for 1 is:
1

Integer partition for 2 is:
2
11

Integer partition for 3 is:
3
12
111

Integer partition for 4 is:
4
13
112
1111
22

Integer partition for 5 is:
5
14
113
1112
11111
122
23

```

The terminal window has a dark background and a light blue border. The output is displayed in a monospaced font. The browser's address bar and tabs are visible at the top. The Windows taskbar is visible at the bottom of the screen.

3. Java program to delete a node from the middle of the singly linked list

In this program, we will create a singly linked list and delete a node from the middle of the list. To accomplish this task, we will calculate the size of the list and then divide it by 2 to get the mid-point of the list. Node temp will point to head node. We will iterate through the list till midpoint is reached. Now, the temp will point to middle node and node current will point to node

previous to temp. We delete the middle node such that current's next node will point to temp's next node.

```
public class deleteMid{

class Node{

int data;

Node next;

public Node(int data)

{

this.data = data;

this.next = null;

}

}

public Node head = null;

public Node tail = null;


public int size;

public void addNode(int data) {

Node newNode = new Node(data);

if(head == null) {

head = newNode;

tail = newNode;

}

else {

tail.next = newNode;

tail = newNode;

}
```

```
size++;  
  
}  
  
void deleteFromMid() {  
  
Node temp, current;  
  
if(head == null) {  
  
System.out.println("List is empty");  
  
return;  
  
}  
  
else {  
  
int count = (size % 2 == 0) ? (size/2) : ((size+1)/2);  
  
if( head != tail ) {  
  
temp = head;  
  
current = null;  
  
for(int i = 0; i < count-1; i++){  
  
current = temp;  
  
temp = temp.next;  
  
}  
  
if(current != null) {  
  
current.next = temp.next;  
  
temp = null;  
  
}  
  
else {  
  
head = tail = temp.next;  
  
temp = null;  
  
}  
  
}
```

```
}  
  
else {  
  
    head = tail = null;  
  
}  
  
}  
  
size--;  
  
}
```

```
public void display() {
```

```
    Node current = head;  
  
    if(head == null) {  
  
        System.out.println("List is empty");  
  
        return;  
  
        }  
  
        while(current != null) {  
  
            System.out.print(current.data + " ");  
  
            current = current.next;  
  
        }  
  
        System.out.println();  
  
    }
```

```
public static void main(String[] args) {
```

```
deleteMid sList = new deleteMid();
```

```
sList.addNode(1);
```

```
sList.addNode(2);
```

```
sList.addNode(3);
```

```
sList.addNode(4);
```

```
System.out.println("Original List: ");
```

```
sList.display();
```

```
while(sList.head != null) {
```

```
    sList.deleteFromMid();
```

```
    System.out.println("Updated List: ");
```

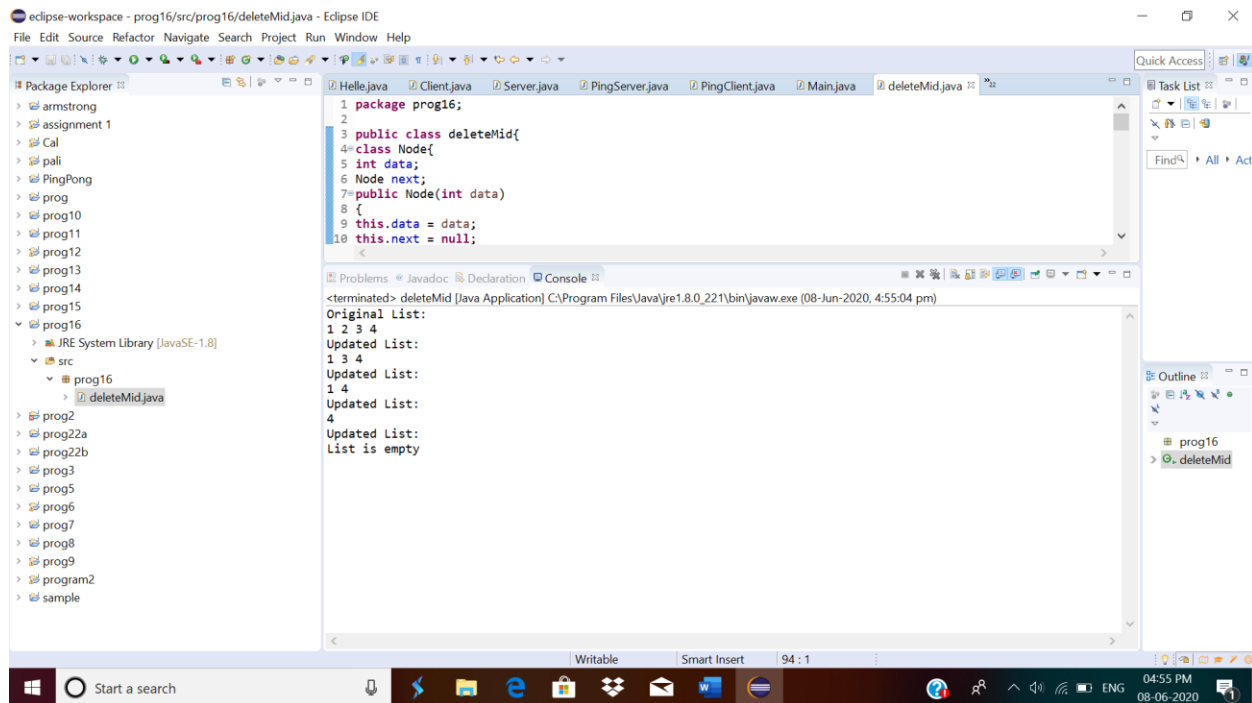
```
    sList.display();
```

```
}
```

```
}
```

```
}
```

Output:



4. Program program to find whether a string is a palindrome or not

Description:

Write a python function that will take a string and checks whether it is a palindrome or not. Return If it a palindrome, print true else print false

Eg: String is : 'aba'

Output: True

```
def isPalindrome(s):
```

```
    return s == s[::-1]
```

```
s = input()
```

```
ans = isPalindrome(s)
```

```
if ans:
```

```
    print("Yes")
```

```
else:
```

```
    print("No")
```

output:



```
Python 3.7.4 Shell
File Edit Shell Debug Options Window Help
Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\hp\Desktop\program\palil.py =====
aba
Yes
>>>
```