

DAILY ONLINE ACTIVITIES SUMMARY

Date:	8-06-2020	Name:	ASHIKA
Sem & Sec	6 A	USN:	4AL17CS016
Online Test Summary			
Subject	CNSC		
Max. Marks	60	Score	40
Certification Course Summary			
Course	Python for data science		
Certificate Provider	Cognitive class	Duration	5 hour
Coding Challenges			
Problem Statement: <div style="margin-left: 40px;"><ol style="list-style-type: none">1. Write C++ program to Check whether a number can be represented as difference of two squares2. C Program to Generate All the Set Partitions of n Numbers Beginning from 1 and so on3. Java program to delete a node from the middle of the singly linked list4. Program program to find whether a string is a palindrome or not</div>			
Status: done(executed)			
Uploaded the report in Github		yes	
If yes Repository name		https://github.com/ASHIKA-05/DAILY-REPORT	

Uploaded the report in slack

yes

Subject : CNSC

techgig.com/challenge/result/mcq/K29yVmJrdVpEd21CaVcxTy9OMU1jdz09

Apps Gmail YouTube Maps Virtual Classroom Virtual Classroom (3) New Messages!

Logout

Results Analytics

Test 3 submitted
MCQ
Your Score
15 / 24

Test 1 submitted
MCQ
Your Score
9 / 16

Test 2 submitted
MCQ
Your Score
16 / 20

CERTIFICATION COURSE

courses.cognitiveclass.ai/courses/course-v1:Cognitiveclass+PY0101EN+v2/courseware/89227024130b43f684d95376901b65c8/aed4761a25e64d57...

Apps Gmail YouTube Maps Virtual Classroom Virtual Classroom (3) New Messages!

Conditions and Branching (10:14)

PY0101EN - Conditions and Branching 10:14

Watch later Share

SPSS IN 5 MIN

STATISTICS BY GROUPS

STATISTICS 101

STATISTICS 101

1990 is greater than 1983 so this condition is also true.

We can verify by examining the corresponding second number line.

In the final number line, the green region indicates where the area is true.

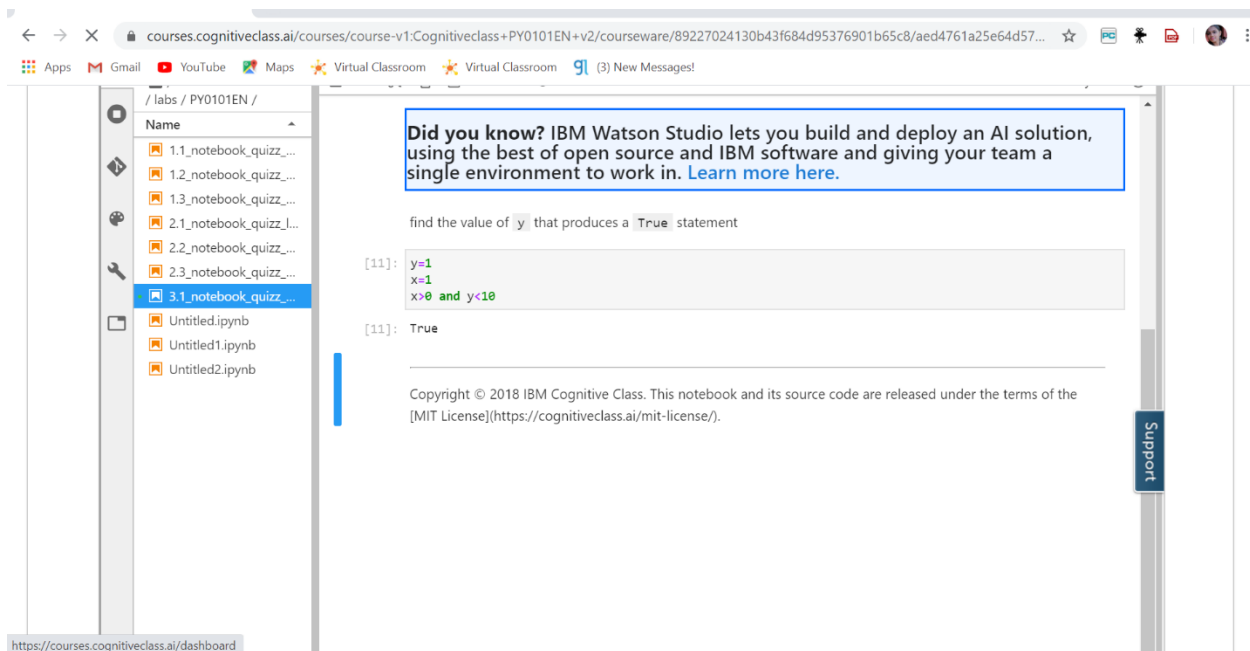
Similarly, this region corresponds to where both statements are true.

We see that 1983 falls in the area.

Therefore we execute the statement.

Branching allows us to run different statements for different inputs.

https://courses.cognitiveclass.ai/dashboard



ONLINE CODEING

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

Given a number N, the task is to check if this number can be represented as the difference of two perfect squares or not.

input: N = 3
 Output: Yes
 Explanation:
 $2^2 - 1^1 = 3$

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
bool difSquare(int n)
```

```
{
```

```
    if (n % 4 != 2) {
```

```
        return true;
    }

    return false;
}

int main()
{

    int n = 3;

    if (difSquare(n)) {
        cout << "Yes\n";
    }

    else {
        cout << "No\n";
    }

    return 0;
}
```

Output:



```
#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:

onlinegdb.com/online_python_compiler

SPONSOR Red Hat — If your organization wasn't already digitally transforming, it has to now.

main.c

```
1 // C program to generate all set partition from 1 to N: 5
2 #include <stdio.h>
3
4 int main()
5 {
6     int N;
7     printf("Enter a number N to generate all set partition from 1 to N: ");
8     scanf("%d", &N);
9
10    // Generate all set partition from 1 to N
11    generatePartition(1, N);
12
13    return 0;
14 }
```

input

```
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
```

Connecting...

onlinegdb.com/online_python_compiler

SPONSOR Red Hat — If your organization wasn't already digitally transforming, it has to now.

main.c

```
1 // C program to generate all set partition from 1 to N: 5
2 #include <stdio.h>
3
4 int main()
5 {
6     int N;
7     printf("Enter a number N to generate all set partition from 1 to N: ");
8     scanf("%d", &N);
9
10    // Generate all set partition from 1 to N
11    generatePartition(1, N);
12
13    return 0;
14 }
```

input

```
Integer partition for 4 is:
4
13
112
1111
22
Integer partition for 5 is:
5
14
113
1112
11111
122
23
...Program finished with exit code 0
Press ENTER to exit console.
```

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:

The screenshot shows a web browser with multiple tabs, including 'online python compiler' and 'Online Java Compiler'. The active page is 'tutorialspoint.com/compile_java_online.php'. The interface features a 'codingground' logo and a title 'Compile and Execute Java Online (JDK 1.8.0)'. It has tabs for 'Execute', 'Share', 'Source File', and 'STDIN'. The main editor contains Java code for deleting a node from a linked list. The 'Result' panel on the right shows the command '\$javac deleteMid.java' and '\$java -Xmx128M -Xms16M deleteMid', followed by the program's output: 'Original List: 1 2 3 4', 'Updated List: 1 3 4', 'Updated List: 1 4', 'Updated List: 4', and 'Updated List: List is empty'. The Windows taskbar at the bottom shows the time as 17:28 on 2020-06-08.

```
65- }
66- while(current != null) {
67-     System.out.print(current.data + " ");
68-     current = current.next;
69- }
70- System.out.println();
71- }
72-
73- public static void main(String[] args) {
74-
75-     deleteMid slist = new deleteMid();
76-
77-     slist.addNode(1);
78-     slist.addNode(2);
79-     slist.addNode(3);
80-     slist.addNode(4);
81-
82-     System.out.println("Original List: ");
83-     slist.display();
84-
85-     while(slist.head != null) {
86-         slist.deleteFromMid();
87-         System.out.println("Updated List: ");
88-         slist.display();
89-     }
90- }
91- }
92-
93- }
```

Establishing secure connection...

Result

```
$javac deleteMid.java
$java -Xmx128M -Xms16M deleteMid

Original List:
1 2 3 4
Updated List:
1 3 4
Updated List:
1 4
Updated List:
4
Updated List:
List is empty
```

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 3.7.6 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.6 (tags/v3.7.6:43364a7ae0, Dec 19 2019, 00:42:30) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python37/rev.py =====
aba
Yes
>>> |
```

Ln: 7 Col: 4

Windows taskbar: Search, File Explorer, Edge, Mail, Calendar, Chrome, Task View, Network, Volume, ENG, 14:28, 08-06-2020