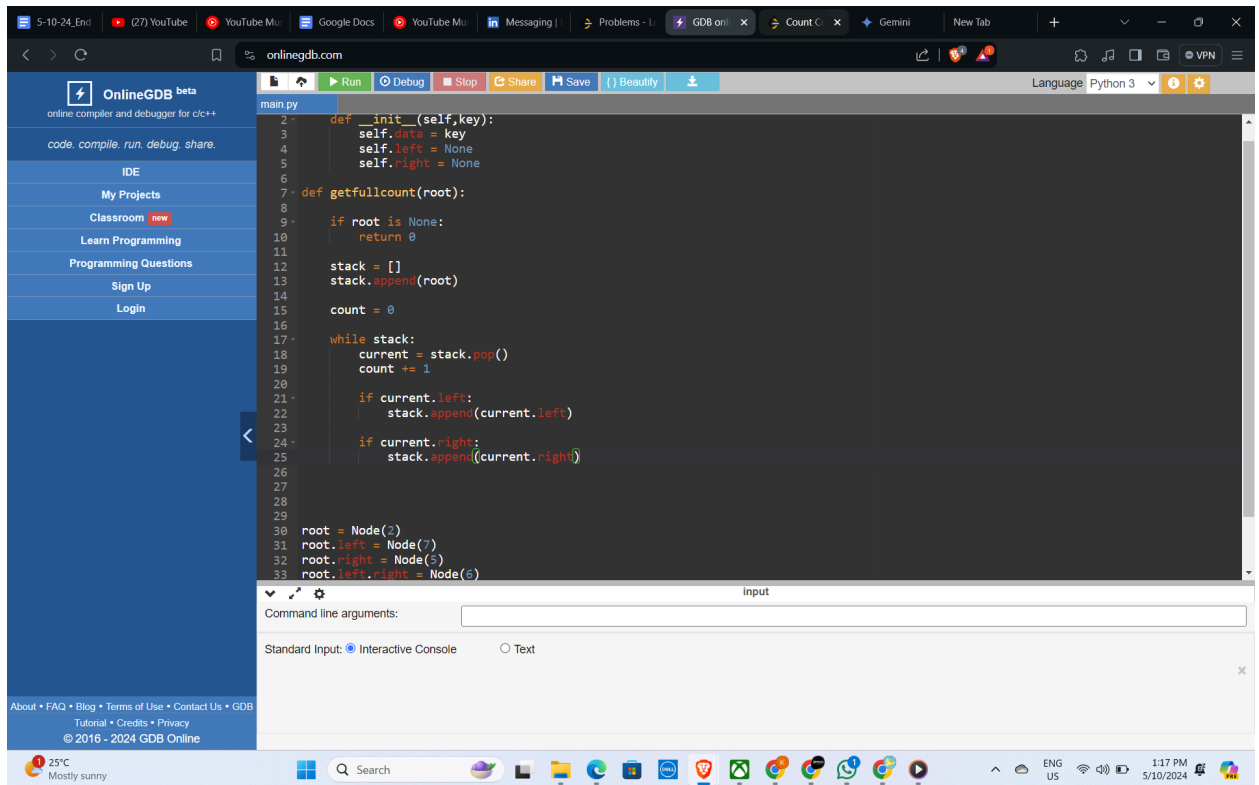


Day95 - May10th 2024

1. Started my day as usual
2. Wrote a program to get full count of binary tree



The screenshot shows the OnlineGDB web interface. The browser tabs include '5-10-24_End', '(27) YouTube', 'YouTube Mu...', 'Google Docs', 'YouTube Mu...', 'Messaging I', 'Problems - I', 'GDB onl...', 'Count C...', 'Gemini', and 'New Tab'. The OnlineGDB interface has a left sidebar with links: 'code, compile, run, debug, share.', 'IDE', 'My Projects', 'Classroom new', 'Learn Programming', 'Programming Questions', 'Sign Up', and 'Login'. The main editor area shows a Python file named 'main.py' with the following code:

```
1 def __init__(self, key):
2     self.data = key
3     self.left = None
4     self.right = None
5
6
7 def getFullcount(root):
8
9     if root is None:
10        return 0
11
12    stack = []
13    stack.append(root)
14
15    count = 0
16
17    while stack:
18        current = stack.pop()
19        count += 1
20
21        if current.left:
22            stack.append(current.left)
23
24        if current.right:
25            stack.append(current.right)
26
27
28
29
30 root = Node(2)
31 root.left = Node(7)
32 root.right = Node(5)
33 root.left.right = Node(6)
```

Below the code editor, there is an 'Input' section with 'Command line arguments:' and a text input field. Below that, 'Standard Input' is set to 'Interactive Console' with a radio button, and 'Text' is also an option. The bottom of the browser shows a Windows taskbar with a search bar, various application icons, and system tray information: '25°C Mostly sunny', 'ENG US', and '1:17 PM 5/10/2024'.

3. Worked on my AWS data engineering project
Pls find the doc here :
https://docs.google.com/document/d/1uJMsg9_PhgtidsLXmi6MaweQkk-q3asVdv36dvFJif4/edit?usp=sharing
4. Ended my day by solving complex SQL question

SQLQuery1.sql - KAUSHI\SQLEXPRESS.master (KAUSHI\jamka (63)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Connect -> KAUSHI\SQLEXPRESS.master (KAUSHI\jamka (63))

Object Explorer: KAUSHI\SQLEXPRESS.master (KAUSHI\jamka (63))

Query: SQLQuery1.sql - K..KAUSHI\jamka (63)

```
--Write a SQL query to find out supplier_id, Product_id, no_of_days and starting_date
--record date for which stock quantity is less than 50 for two or more consecutive days

select * from stock;
```

Results: 19 rows

supplier_id	product_id	stock_quantity	record_date
1	1	60	2022-01-01
1	1	40	2022-01-02
1	1	35	2022-01-03
1	1	45	2022-01-04
1	1	51	2022-01-06
1	1	55	2022-01-09
1	1	25	2022-01-10
1	1	45	2022-01-11
1	1	45	2022-01-15
1	1	38	2022-01-16
1	2	45	2022-01-08
1	2	40	2022-01-09
2	1	45	2022-01-06
2	1	55	2022-01-07
2	2	45	2022-01-08
2	2	48	2022-01-09
2	2	35	2022-01-10
2	2	52	2022-01-15
2	2	23	2022-01-16

Query executed successfully.

SQLQuery1.sql - KAUSHI\SQLEXPRESS.master (KAUSHI\jamka (63)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Connect -> KAUSHI\SQLEXPRESS.master (KAUSHI\jamka (63))

Object Explorer: KAUSHI\SQLEXPRESS.master (KAUSHI\jamka (63))

Query: SQLQuery1.sql - K..KAUSHI\jamka (63)

```
with cte as(
select supplier_id,product_id,record_date,lag(record_date,1,record_date) over(partition by supplier_id,product_id
order by record_date) as prev_record_date,
DATEDIFF(day,lag(record_date,1,record_date) over(partition by supplier_id,product_id
order by record_date),record_date) as days_diff
from stock
where stock_quantity < 50),
cte2 as(
select *,case when days_diff <= 1 then 0 else 1 end as group_flag,
sum(case when days_diff <= 1 then 0 else 1 end) over(partition by supplier_id,product_id
order by record_date) as group_id
from cte)
select supplier_id,product_id,count(*) as no_of_records,min(record_date) as first_date
from cte2
group by supplier_id,product_id,group_id
```

Results: 7 rows

supplier_id	product_id	no_of_records	first_date
1	1	3	2022-01-02
1	1	2	2022-01-10
1	1	2	2022-01-15
1	2	2	2022-01-08
2	1	1	2022-01-06
2	2	3	2022-01-08
2	2	1	2022-01-16

Query executed successfully.

/*Explanation:

Step1 : first we have used lag to find the datediff between dates where stock_quantity is less than 50

Step2 : Now using case when we'll assign 0 if the diff is less than 1 else 1 and if we use running sum on group_flag with partition by supplier,product then we will get the column on which we can apply the group by

Step3 : Now we groupby on supplier,product and group_id to get the required result