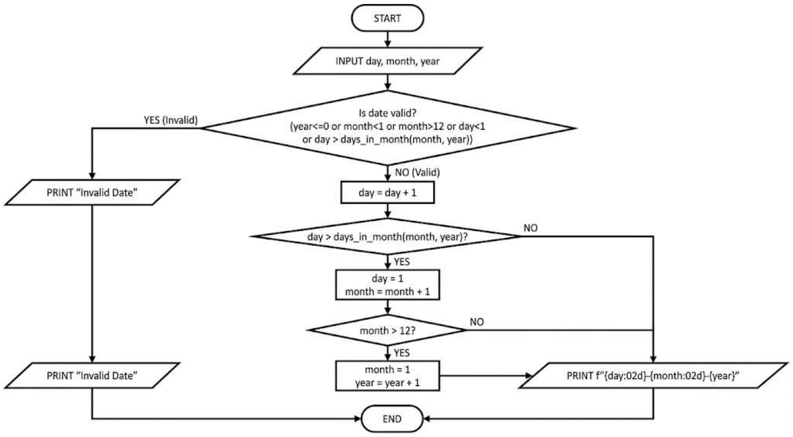
Name:Aryan Shegokar PRN:25070521180

6.1.1. Incremented Date

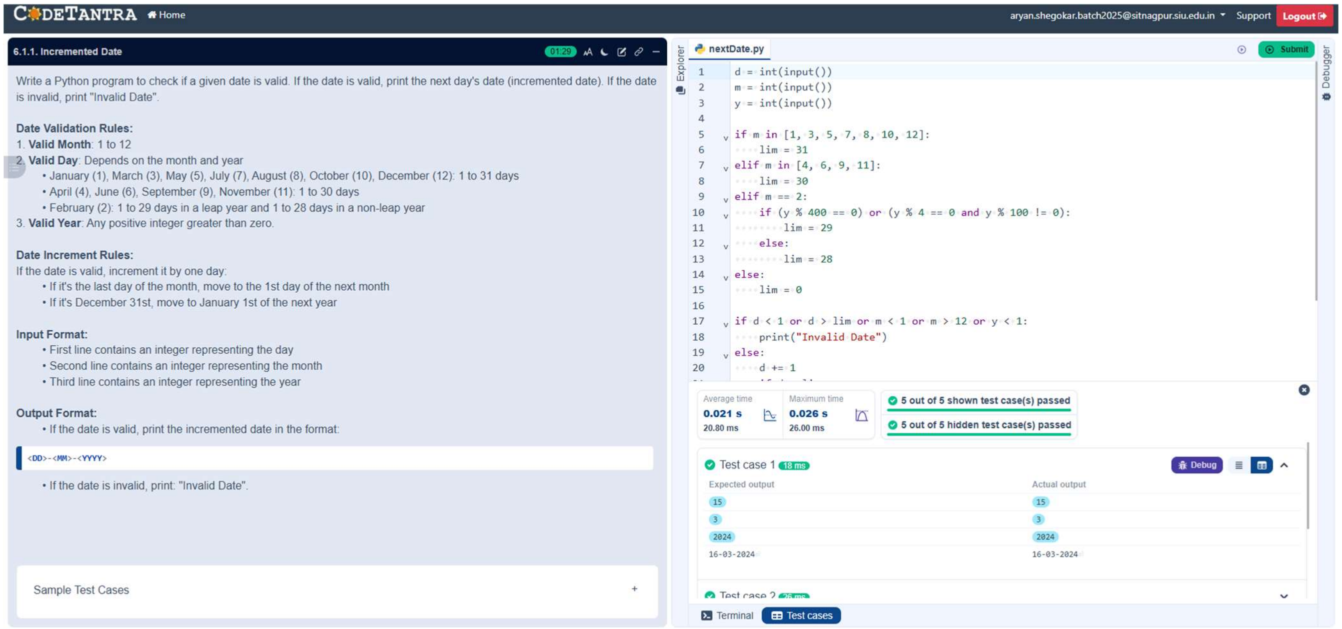
Algorithm: Short Next Date

- **Step 1:** Start.
- **Step 2:** Input **d** (day), **m** (month), **y** (year).
- **Step 3:** Determine the day limit (**lim**):
    - If **m** is 1, 3, 5, 7, 8, 10, or 12, set **lim** to 31.
    - If **m** is 4, 6, 9, or 11, set **lim** to 30.
    - If **m** is 2, check for leap year:
        - If yes, set **lim** to 29.
        - If no, set **lim** to 28.
    - Otherwise, set **lim** to 0.
- **Step 4:** Check Validity:
    - If **d** is out of range (less than 1 or greater than **lim**), or **m** is invalid (not 1-12), or **y** is invalid (< 1):
    - Print "Invalid Date" and stop.
- **Step 5:** Calculate Next Date:
    - Increment **d** by 1.
    - If **d** > **lim**:
        - Reset **d** to 1.
        - Increment **m** by 1.
        - If **m** > 12:
            - Reset **m** to 1.
            - Increment **y** by 1.
- **Step 6:** Print the result in DD-MM-YYYY format.
- **Step 7:** End.

Flow chart:



Execution



6.1.1. Incremented Date

Write a Python program to check if a given date is valid. If the date is valid, print the next day's date (incremented date). If the date is invalid, print "Invalid Date".

Date Validation Rules:

1. **Valid Month**: 1 to 12
2. **Valid Day**: Depends on the month and year
   - January (1), March (3), May (5), July (7), August (8), October (10), December (12): 1 to 31 days
   - April (4), June (6), September (9), November (11): 1 to 30 days
   - February (2): 1 to 29 days in a leap year and 1 to 28 days in a non-leap year
3. **Valid Year**: Any positive integer greater than zero.

Date Increment Rules:

If the date is valid, increment it by one day:

- If it's the last day of the month, move to the 1st day of the next month
- If it's December 31st, move to January 1st of the next year

Input Format:

- First line contains an integer representing the day
- Second line contains an integer representing the month
- Third line contains an integer representing the year

Output Format:

- If the date is valid, print the incremented date in the format:

```
<DD>-<MM>-<YYYY>
```

- If the date is invalid, print: "Invalid Date".

Sample Test Cases +

```python
d = int(input())
m = int(input())
y = int(input())

if m in [1, 3, 5, 7, 8, 10, 12]:
    lim = 31
elif m in [4, 6, 9, 11]:
    lim = 30
elif m == 2:
    if (y % 400 == 0) or (y % 4 == 0 and y % 100 != 0):
        lim = 29
    else:
        lim = 28
else:
    lim = 0

if d < 1 or d > lim or m < 1 or m > 12 or y < 1:
    print("Invalid Date")
else:
    d += 1
```

Average time 0.021 s 20.80 ms

Maximum time 0.026 s 26.00 ms

5 out of 5 shown test case(s) passed

5 out of 5 hidden test case(s) passed

Test case 1 18 ms

Debug

Expected output	Actual output
15	15
3	3
2024	2024
16-03-2024	16-03-2024

Test case 2

Terminal Test cases