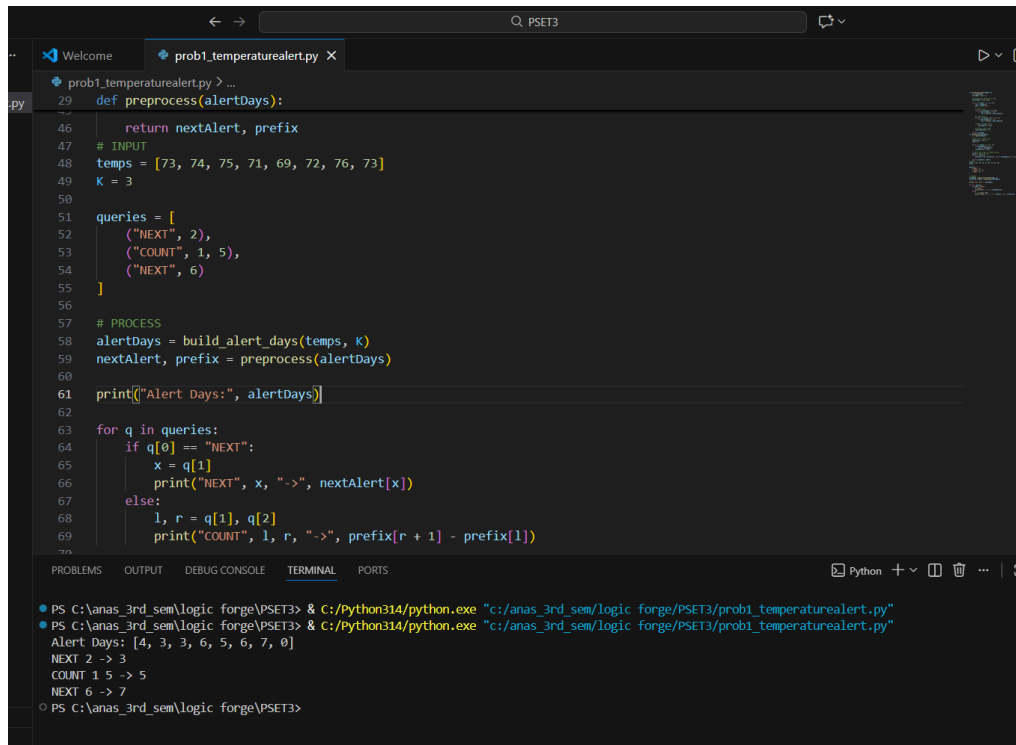


## Pset 3 outputs

### Q1

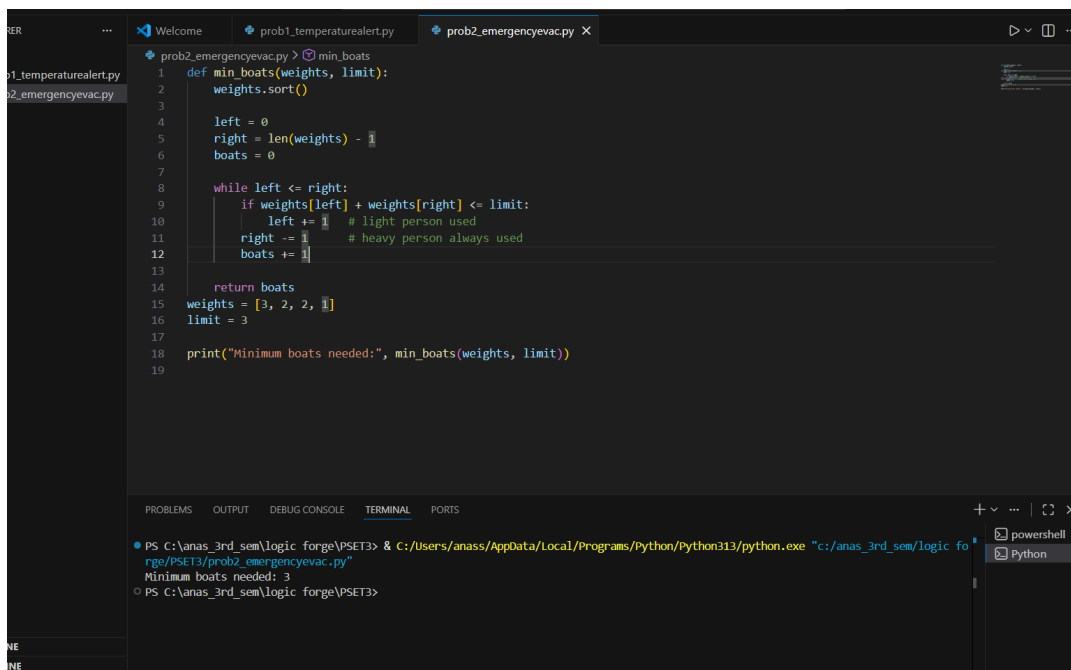


```
def preprocess(alertDays):
    return nextAlert, prefix
# INPUT
temps = [73, 74, 75, 71, 69, 72, 76, 73]
K = 3
queries = [
    ("NEXT", 2),
    ("COUNT", 1, 5),
    ("NEXT", 6)
]
# PROCESS
alertDays = build_alert_days(temps, K)
nextAlert, prefix = preprocess(alertDays)
print("Alert Days:", alertDays)
for q in queries:
    if q[0] == "NEXT":
        x = q[1]
        print("NEXT", x, "->", nextAlert[x])
    else:
        l, r = q[1], q[2]
        print("COUNT", l, r, "->", prefix[r + 1] - prefix[l])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\anas_3rd_sem\logic forge\PSET3> & C:/Python314/python.exe "c:/anas_3rd_sem/logic forge/PSET3/prob1_temperaturealert.py"
PS C:\anas_3rd_sem\logic forge\PSET3> & C:/Python314/python.exe "c:/anas_3rd_sem/logic forge/PSET3/prob1_temperaturealert.py"
Alert Days: [4, 3, 3, 6, 5, 6, 7, 0]
NEXT 2 -> 3
COUNT 1 5 -> 5
NEXT 6 -> 7
PS C:\anas_3rd_sem\logic forge\PSET3>
```

### Q2



```
def min_boats(weights, limit):
    weights.sort()
    left = 0
    right = len(weights) - 1
    boats = 0
    while left <= right:
        if weights[left] + weights[right] <= limit:
            left += 1
            right -= 1
        else:
            right -= 1
        boats += 1
    return boats
weights = [3, 2, 2, 1]
limit = 3
print("Minimum boats needed:", min_boats(weights, limit))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\anas_3rd_sem\logic forge\PSET3> & C:/Users/anass/AppData/Local/Programs/Python/Python313/python.exe "c:/anas_3rd_sem/logic fo
rg/PSET3/prob2_emergencyevac.py"
Minimum boats needed: 3
PS C:\anas_3rd_sem\logic forge\PSET3>
```

Q3

```

1 def broadcast_network_feed(N, Q, K, operations):
2     for mid, t, sender, critical in reversed(messages):
3         if sender == u or sender in subscriptions[u]:
4             if mid in user_messages[sender]:
5                 feed.append((t, critical, mid))
6             if len(feed) == 10:
7                 break
8
9         if not feed:
10            print("EMPTY")
11        else:
12            feed.sort(key=lambda x: (-x[0], -x[1]))
13            print(" ".join(str(x[2]) for x in feed))
14
15 N, Q, K = 3,9,2
16 operations = [
17     "S 1 2",
18     "S 1 3",
19     "B 2 5",
20     "B 3 9",
21 ]
22
23 if __name__ == "__main__":
24     main()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

F 1
F 2
2 1
3 2
1
PS C:\anas_3rd_sem\logic forge\PSET3> & C:/Python314/python.exe "c:/anas_3rd_sem/logic forge/PSET3/prob3_broadcastnetwork.py"
2 1
3 2
1
[]

```

Q4

```

1 def find_anagrams(s, p):
2     freq_window = [0] * 26
3     freq_p = [0] * 26
4     for i in range(len(s)):
5         freq_window[ord(s[i]) - ord('a')] += 1
6         if i >= len(p) - 1:
7             freq_p[ord(p[i]) - ord('a')] += 1
8             if freq_window == freq_p:
9                 result.append(i - window_size + 1)
10            freq_p = [0] * 26
11    return result
12
13 def main():
14     s = "cbaebabacd"
15     p = "abc"
16
17     ans = find_anagrams(s, p)
18     print(ans)
19
20 if __name__ == "__main__":
21     main()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

2 1
3 2
1
PS C:\anas_3rd_sem\logic forge\PSET3> & C:/Python314/python.exe "c:/anas_3rd_sem/logic forge/PSET3/prob3_broadcastnetwork.py"
2 1
3 2
1
Traceback (most recent call last):
  File "c:/anas_3rd_sem/logic forge\PSET3/prob3_broadcastnetwork.py", line 70, in <module>
    operations.append(input().strip())
    ~~~~~^~~~~~
KeyboardInterrupt
PS C:\anas_3rd_sem\logic forge\PSET3> & C:/Python314/python.exe "c:/anas_3rd_sem/logic forge/PSET3/prob4_scrambledkeyword.py"
[0, 6]
PS C:\anas_3rd_sem\logic forge\PSET3>

```

## Q5

```

prob5_longestmirror.py > main
1 def longest_mirrored_phrase(s):
2     n = len(s)
3
4     # dp[i][j] = LPS length from i to j
5     dp = [[0] * n for _ in range(n)]
6
7     # single characters
8     for i in range(n):
9         dp[i][i] = 1
10
11    # build table bottom-up
12    for length in range(2, n + 1):
13        for i in range(n - length + 1):
14            j = i + length - 1
15
16            if s[i] == s[j]:
17                if length == 2:
18                    dp[i][j] = 2
19                else:
20                    dp[i][j] = 2 + dp[i + 1][j - 1]
21            else:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\anas_3rd_sem\logic forge\PSET3> & C:/Python314/python.exe "c:/anas_3rd_sem/logic forge/PSET3/prob5_longestmirror.py"
4
PS C:\anas_3rd_sem\logic forge\PSET3>

```

## Q6

```

prob6_suspicious_device.py > find_suspicious_device
1 def find_suspicious_device(nums):
2     freq = {}
3     n = len(nums) // 2
4
5     for x in nums:
6         freq[x] = freq.get(x, 0) + 1
7         if freq[x] == n:
8             return x
9
10    def main():
11        # HARD-CODED INPUT
12        nums = [2, 1, 2, 5, 3, 2]
13
14        print(find_suspicious_device(nums))
15
16    if __name__ == "__main__":
17        main()
18

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

• :/anas_3rd_sem/logic forge/PSET3/prob5_longestmirror.py"
4
PS C:\anas_3rd_sem\logic forge\PSET3> ^C
PS C:\anas_3rd_sem\logic forge\PSET3> & C:/Python314/python.exe "c:/anas_3rd_sem/logic forge/PSET3/prob6_suspicious_device.py"
2
PS C:\anas_3rd_sem\logic forge\PSET3>

```