

Problem1:

The screenshot shows a Windows desktop environment with a Visual Studio Code (VS Code) window open. The title bar reads "File Edit Selection View Go Run Terminal Help" and "Q Logic-Forge-PSET2". The left sidebar is titled "EXPLORER" and lists several files under "LOGIC-FORGE-PSET2": Challenge1_MountainPeak.py, Challenge2_BalancedScales.py, Challenge3_MirrorQuest.py, Challenge4_RoyalTreasury.py, Challenge5_MasterScheduler'sDilemma.py, Challenge6_High-StakesFreelancer.py, Challenge7_KokoStanathanBuffet.py, and tempCodeRunnerfile.py. The main editor area contains Python code for calculating the number of ways to reach a mountain peak. The code uses dynamic programming to calculate the number of ways to sum up to n steps. It includes comments explaining the logic and prints the result for n=4 and n=3. The bottom status bar shows the path "PS C:\Users\VRComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Logic-Forge-PSET2\Challenge1_MountainPeak.py", the file name "Challenge1_MountainPeak.py", and the status "Running".

```
file:///C:/Users/VRComputers/Desktop/VS%20code%20stuff/Logic-Forge-PSET2/Logic-Forge-PSET2/tempCodeRunnerfile.py
1 import time
2
3 def count_ways_to_summit(n):
4     if n <= 1:
5         return n
6
7     #For two ways: 1 step or 2 steps
8     first = 1
9     second = 1
10
11    #Start from step 2 to n
12    for _ in range(2, n + 1):
13        current = first + second
14        first = second
15        second = current
16
17    return second
18
19
20 n = int(input("Enter the number of steps to the mountain peak: "))
21
22 start_time = time.perf_counter()
23 ways = count_ways_to_summit(n)
24 end_time = time.perf_counter()
25
26 elapsed_time = end_time - start_time
27
28 #It asked for number of ways and not the actual paths.
29 print(f"For {n} steps, there are {ways} distinct ways to reach the mountain peak.")
30 print(f"Time taken to compute: {elapsed_time:.10f} seconds")
31
```

Problem2:

The screenshot shows a VS Code interface with the following details:

- File Explorer:** Shows a tree view of files under "LOGIC-FORGE-PSET2". The file "Challenge2_BalancedScales.py" is open in the editor.
- Editor:** Displays the Python code for "Challenge2_BalancedScales.py". The code implements a solution to the "Balanced Scales" problem using a set-based approach to find all possible sums of elements in the array.
- Terminal:** Shows command-line output for three test cases (Arr1, Arr2, Arr3) and a final command to run the script.
- Performance Monitor:** A sidebar on the right displays system performance metrics like GPU Utilization, CPU Temperature, and CPU Power.

```
File Edit Selection View Go Run Terminal Help ↵ → Q Logic-Forge-PSET2

EXPLORER ... nge3_MirrorQuest.py Challenge4_RoyalTreasury.py Challenge5_MasterScheduler'sDilemma.py Challenge6_High-StakesFreelancer.py Challenge1_MountainPeak.py Challenge2_BalancedScales.py

Challenge1_MountainPeak.py
Challenge2_BalancedScales.py
Challenge3_MirrorQuest.py
Challenge4_RoyalTreasury.py
Challenge5_MasterScheduler'sDilemma.py
Challenge6_High-StakesFreelancer.py
Challenge7_KokoBananaBuffet.py
tempCodeRunnerFile.py U

1 import time
2
3 def can_balance_scales(arr):
4
5     total_sum = sum(arr)
6
7     #cont split if odd
8     if total_sum % 2 != 0:
9         return False
10
11     target = total_sum // 2
12
13     ## start with 0.
14     possible_sums = {0}
15
16     for numbers in arr:
17
18         current_sums = set()
19
20         for sumsum in possible_sums:
21
22             new_sum = sumsum + numbers
23
24             if new_sum <= target:
25                 current_sums.add(new_sum)
26
27             #Add the new sum to the collection of possible sums
28             possible_sums.update(current_sums)
29
30     if target in possible_sums:
31         return True
32
33     return False
34

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\VRComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Logic-Forge-PSET2> python -u "c:\Users\VRComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Logic-Forge-PSET2\Challenge2_BalancedScales.py"
Arr1: True
Time taken to compute: 0.0008293000 seconds

Arr2: False
Time taken to compute: 0.0000439000 seconds

Arr3: True
Time taken to compute: 0.0001189000 seconds

PS C:\Users\VRComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Logic-Forge-PSET2>

GPU UTIL   0% GPU PWR   4 W GPU TEMP   0 mV GPU VOLTAGE   0.0V GPU MEM UTIL   0.026 MB CPU UTIL   0.5% CPU VOLTAGE   1.5 V CPU TEMP   36.3 °C CPU POWER   16.7W
```

Problem3:

```
def find_longest_mirror_length(s):
    n = len(s)
    dp = [[0] * n for _ in range(n)]
    #Base case: when there is only one character
    for i in range(n):
        dp[i][i] = 1
    #Build substrings of length 2 to n
    for length in range(2, n + 1):
        for i in range(n - length + 1):
            j = i + length - 1
            if s[i] == s[j]: #Characters match
                dp[i][j] = 2 + dp[i + 1][j - 1]
            else: #Characters don't match
                dp[i][j] = max(dp[i + 1][j], dp[i][j - 1])
    return dp[0][n - 1]

s1 = "bbbab"
print(f"Length: {find_longest_mirror_length(s1)}")
s2 = "cbbd"
print(f"Length: {find_longest_mirror_length(s2)}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\VarComputers\Desktop\VS code stuff\Logic-Forge-PSET2> python -u "c:/Users\VarComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Logic-Forge-PSET2\Challenge3_MirrorQuest.py"

Length: 2

PS C:\Users\VarComputers\Desktop\VS code stuff\Logic-Forge-PSET2>

Code F11 GPU UTIL 4 % GPU PWR 5 W GPU TEMP 31 °C GPU VOLTAGE 781 mV GPU MEM UTIL 945 MB CPU UTIL 9 % CPU VOLTAGE 1.2 V CPU TEMP 34.2 °C CPU POWER 8.5 W SYSTEM MEM UTIL 8.5 GB 8.5 Go Live Profiler

Problem4:

```
def count_payment_combinations(coins, total_sum):
    ways = [0] * (total_sum + 1)
    #Base Case: Choose no coins
    ways[0] = 1
    for coin in coins:
        for amount in range(coin, total_sum + 1):
            ways[amount] += ways[amount - coin]
    return ways[total_sum]

print(f"Case A: {count_payment_combinations([1, 2, 3, 4], 4)}")
print(f"Case B: {count_payment_combinations([2, 5, 3, 6], 10)}")
print(f"Case C: {count_payment_combinations([1, 5], 5)}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\VarComputers\Desktop\VS code stuff\Logic-Forge-PSET2> python -u "c:/Users\VarComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Logic-Forge-PSET2\Challenge4_RoyalTreasury.py"

Case A: 4

Case B: 10

Case C: 8

PS C:\Users\VarComputers\Desktop\VS code stuff\Logic-Forge-PSET2>

Code F11 GPU UTIL 0 % GPU PWR 5 W GPU TEMP 30 °C GPU VOLTAGE 781 mV GPU MEM UTIL 955 MB CPU UTIL 17 % CPU VOLTAGE 1.2 V CPU TEMP 35.6 °C CPU POWER 8.1 W SYSTEM MEM UTIL 8.1 GB 8.1 Go Live Profiler

Problem5:

The screenshot shows a Microsoft Visual Studio Code interface with the following details:

- File Explorer:** Shows a folder named "LOGIC-FORGE-PSET2" containing several Python files related to challenges like "Challenge1_MountainPeak.py", "Challenge2_BalancedScales.py", etc.
- Code Editor:** The main editor window displays a Python script titled "Challenge5_MasterScheduler'sDilemma.py". The code defines a function `min_cancelled_bookings` that sorts intervals by ending time and counts bookings that overlap with them. It then tests this function with three sets of intervals and prints the results.
- Terminal:** The terminal window shows the output of running the script with the command `python -u "C:\Users\VKComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Challenge5_MasterScheduler'sDilemma.py"`. The output indicates:
 - Number of cancelled bookings for interval1 = 1
 - Number of cancelled bookings for interval2 = 2
 - Number of cancelled bookings for interval3 = 0
- Performance Monitor:** A sidebar on the right provides real-time system monitoring for CPU, GPU, and memory usage.
- Bottom Status Bar:** Shows file paths, line numbers, and other status information.

Problem6:

The screenshot shows a Jupyter Notebook interface with the following details:

- File Edit Selection View Go Run Terminal Help** menu bar.
- EXPLORER** sidebar with a tree view of files:
 - LOGIC-FORGE-PSET2
 - Challenge1_MountainPeak.py
 - Challenge2_BalancedScales.py
 - Challenge3_MirrorQuest.py
 - Challenge4_RoyalTreasury.py
 - Challenge5_MasterSchedulersDilemma.py
 - Challenge6_High-StakesFreelancer.py
 - Challenge7_KokoBananaBunfet.py
 - tempCodeRunnerfile.py
- CELL** tab selected in the top right.
- Code** cell content:

```
def maximize_freelance_profit(deadlines, profits):  
    #combines deadlines and profits  
    jobs = list(zip(deadlines, profits))  
    max_deadline = max(deadlines)  
  
    #Sorts the jobs by their profits in descending  
    jobs.sort(key=lambda x: x[1], reverse=True)  
  
    time_slots = [False] * (max_deadline + 1)  
  
    total_jobs = 0  
    total_profit = 0  
  
    for deadline, profit in jobs:  
        for t in range(deadline, 0, -1):  
            if not time_slots[t]: #if slot free  
                time_slots[t] = True #mark slot as occupied  
                total_profit += profit #add profit  
                total_jobs += 1  
                break  
  
    return [total_jobs, total_profit]  
  
#Example 1  
deadlines1 = [4, 1, 1, 1]  
profits1 = [20, 10, 40, 30]  
print(f"Example 1 Output: {maximize_freelance_profit(deadlines1, profits1)}")  
  
#Example 2  
deadlines2 = [2, 1, 2, 1, 1]  
profits2 = [100, 19, 27, 15, 1]  
print(f"Example 2 Output: {maximize_freelance_profit(deadlines2, profits2)}")
```
- PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS** tabs at the bottom.
- TERMINAL** tab content:

```
PS C:\Users\VAComputers\Desktop\VS code stuff\Logic-Forge-PSET2> python -u "C:\Users\VAComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Challenge6_High-StakesFreelancer.py"  
Example 1 Output: [4, 60]  
Example 2 Output: [5, 151]
```
- OUTPUT** tab content:

```
% PS C:\Users\VAComputers\Desktop\VS code stuff\Logic-Forge-PSET2> Logic-Forge-PSET2>
```
- STATUS** bar at the bottom showing: 10 main*, 0.11 s, 0 g, 0 △, AhomRB (3 minutes ago), Ln 12 Col 19, Spaces: 4, UTF-8, Python 3.7.6, SYSTEM MEM UTIL: 8.9 Gc Live, 8.3 Gc Free.

Problem7:

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "LOGIC-FORGE-PSET2". The file "isury.py" is open, containing Python code for calculating minimum speed.
- Code Editor:** Displays the code for "Challenge7_Koko'sBananaBuffet.py". The code implements a binary search algorithm to find the minimum speed required to eat all piles of bananas within a given time limit.
- Terminal:** Shows the command-line output of running the code with example inputs and outputs.
- Performance Monitor:** A floating window displays GPU and CPU performance metrics.

```
def calculate_minimum_speed(piles, k):
    left = 0
    right = max(piles)
    answer = right

    while left <= right:
        mid = (left + right) // 2
        hours = 0
        for pile in piles:
            hours += (pile + mid - 1) // mid #ceiling division

        if hours <= k:
            answer = mid #speed possible
            right = mid - 1 #try slowing down
        else:
            left = mid + 1 #need more speed

    return answer

#Example 1
piles1 = [5, 10, 3]
k1 = 4
print(f"Speed: {calculate_minimum_speed(piles1, k1)}")

#Example 2
piles2 = [5, 10, 15, 20]
k2 = 7
print(f"Speed: {calculate_minimum_speed(piles2, k2)}")
```

```
PS C:\Users\VARComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Logic-Forge-PSET2> python -u "c:\Users\VARComputers\Desktop\VS code stuff\Logic-Forge-PSET2\Challenge7_Koko'sBananaBuffet.py"
Speed: 5
Speed: 10
```

GPU	N/A
GPU UTIL	5 %
GPU PWR	8 W
GPU TEMP	31 °C
GPU VOLTAGE	781 mV
GPU MEM UTIL	961 MB
CPU UTIL	7 %
CPU VOLTAGE	1.3 V
CPU TEMP	37.5 °C
CPU POWER	9.2 W
SYSTEM MEM UTIL	8.4 GB