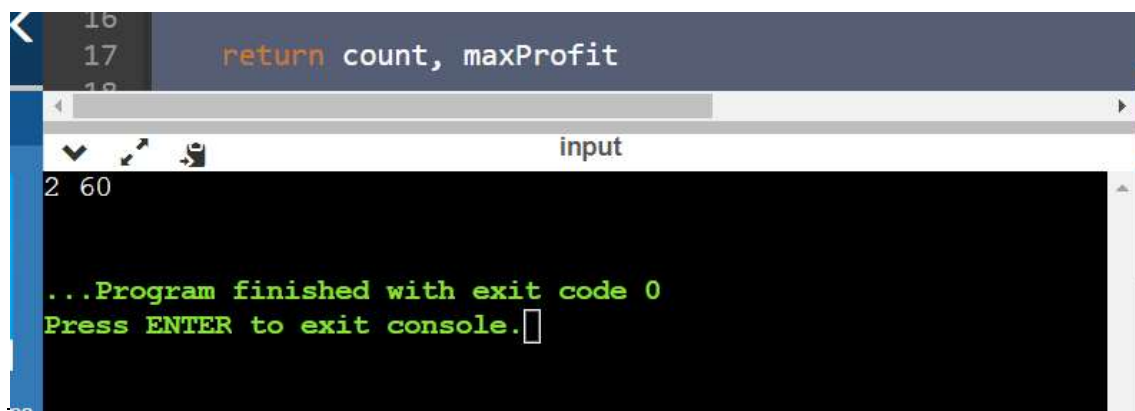


Problem 3: You are given a set of N jobs where each job comes with a **deadline** and **profit**. The profit can only be earned upon completing the job within its deadline. Find the **number of jobs** done and the **maximum profit** that can be obtained. Each job takes a **single unit** of time and only **one job** can be performed at a time.

```
def findMaxProfit(N, jobs):  
    jobs.sort(key=lambda x: x[2], reverse=True)  
    maxDeadline = max(jobs, key=lambda x: x[1])[1]  
    slot = [-1] * maxDeadline  
    count = 0  
    maxProfit = 0  
  
    for job in jobs:  
        for i in range(job[1] - 1, -1, -1):  
            if slot[i] == -1:  
                slot[i] = job[0]  
                count += 1  
                maxProfit += job[2]  
                break  
    return count, maxProfit  
  
jobs = [(1, 4, 20), (2, 1, 10), (3, 1, 40), (4, 1, 30)]  
N = 4  
count, maxProfit = findMaxProfit(N, jobs)  
print(count, maxProfit)
```



The image shows a code editor window with a dark theme. The code editor displays the function `findMaxProfit` and the main execution code. The terminal window below the code editor shows the output of the program, which is `2 60`. The terminal also displays the message `...Program finished with exit code 0` and `Press ENTER to exit console.`

```
16  
17     return count, maxProfit  
18  
input  
2 60  
...Program finished with exit code 0  
Press ENTER to exit console.
```