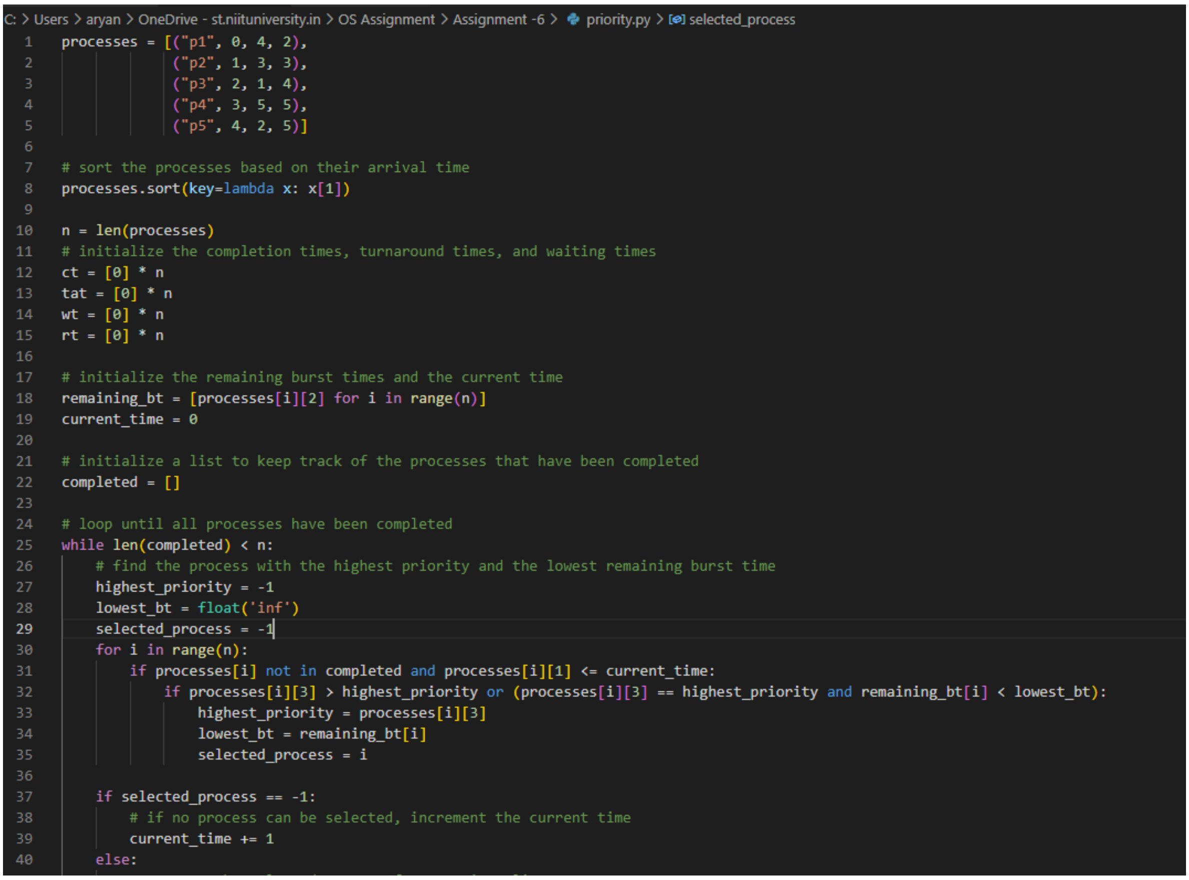
Name – Aryan Sharma

BT21GCS161- B3

Assignment -06

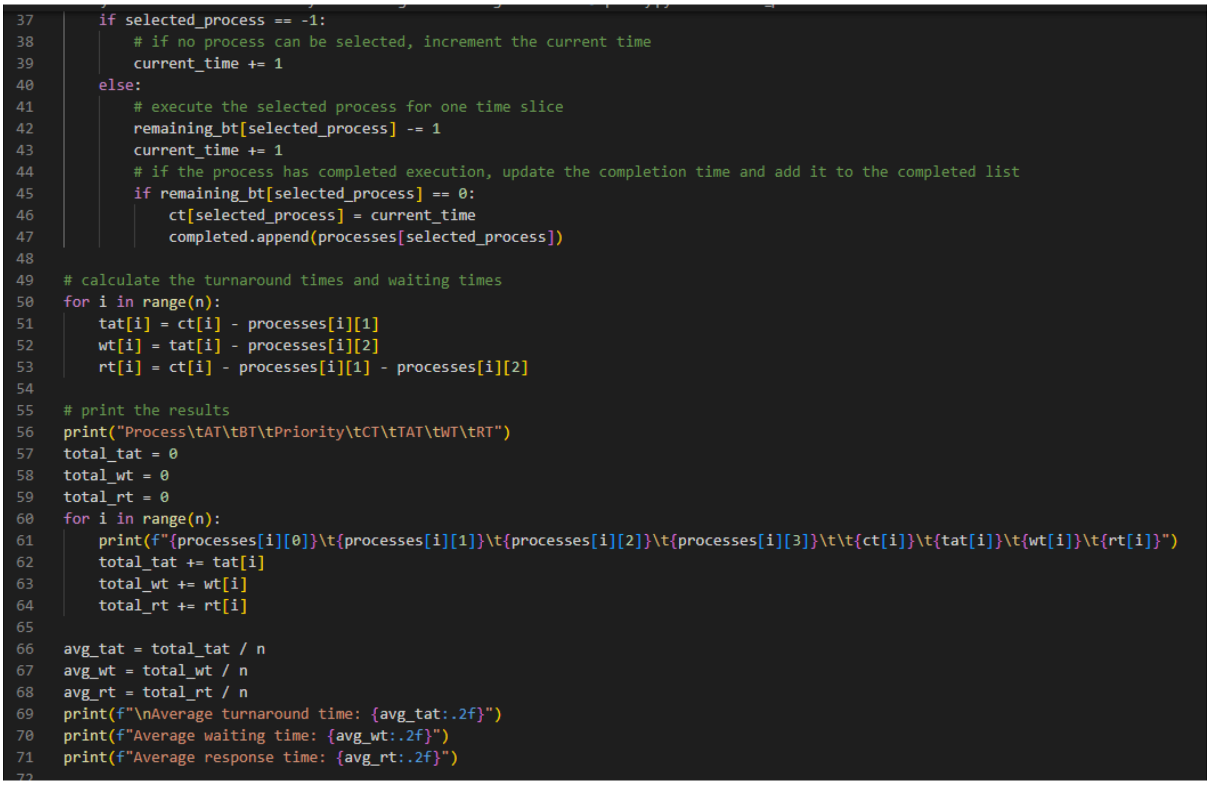
Question – WAP to simulate the pre-emptive priority-based scheduling.

Code –



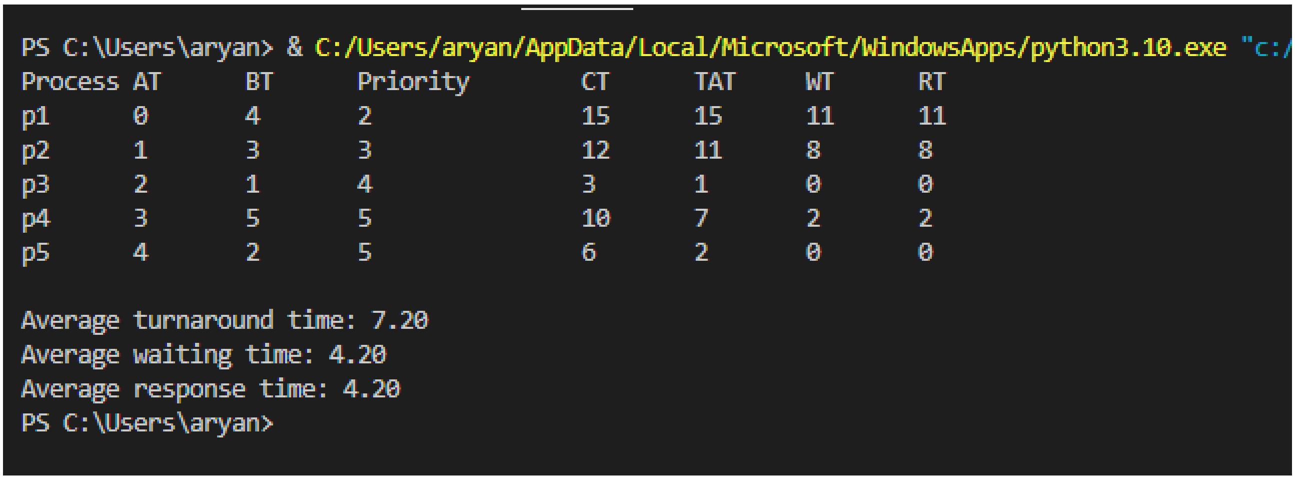
C: > Users > aryan > OneDrive - st.niituniversity.in > OS Assignment > Assignment -6 > priority.py > selected_process

```python
processes = [("p1", 0, 4, 2),
             ("p2", 1, 3, 3),
             ("p3", 2, 1, 4),
             ("p4", 3, 5, 5),
             ("p5", 4, 2, 5)]

# sort the processes based on their arrival time
processes.sort(key=lambda x: x[1])

n = len(processes)
# initialize the completion times, turnaround times, and waiting times
ct = [0] * n
tat = [0] * n
wt = [0] * n
rt = [0] * n

# initialize the remaining burst times and the current time
remaining_bt = [processes[i][2] for i in range(n)]
current_time = 0

# initialize a list to keep track of the processes that have been completed
completed = []

# loop until all processes have been completed
while len(completed) < n:
    # find the process with the highest priority and the lowest remaining burst time
    highest_priority = -1
    lowest_bt = float('inf')
    selected_process = -1
    for i in range(n):
        if processes[i] not in completed and processes[i][1] <= current_time:
            if processes[i][3] > highest_priority or (processes[i][3] == highest_priority and remaining_bt[i] < lowest_bt):
                highest_priority = processes[i][3]
                lowest_bt = remaining_bt[i]
                selected_process = i

    if selected_process == -1:
        # if no process can be selected, increment the current time
        current_time += 1
    else:
```



```python
    if selected_process == -1:
        # if no process can be selected, increment the current time
        current_time += 1
    else:
        # execute the selected process for one time slice
        remaining_bt[selected_process] -= 1
        current_time += 1
        # if the process has completed execution, update the completion time and add it to the completed list
        if remaining_bt[selected_process] == 0:
            ct[selected_process] = current_time
            completed.append(processes[selected_process])

# calculate the turnaround times and waiting times
for i in range(n):
    tat[i] = ct[i] - processes[i][1]
    wt[i] = tat[i] - processes[i][2]
    rt[i] = ct[i] - processes[i][1] - processes[i][2]

# print the results
print("Process\tAT\tBT\tPriority\tCT\tTAT\tWT\tRT")
total_tat = 0
total_wt = 0
total_rt = 0
for i in range(n):
    print(f"{processes[i][0]}\t{processes[i][1]}\t{processes[i][2]}\t{processes[i][3]}\t\t{ct[i]}\t{tat[i]}\t{wt[i]}\t{rt[i]}")
    total_tat += tat[i]
    total_wt += wt[i]
    total_rt += rt[i]

avg_tat = total_tat / n
avg_wt = total_wt / n
avg_rt = total_rt / n
print(f"\nAverage turnaround time: {avg_tat:.2f}")
print(f"Average waiting time: {avg_wt:.2f}")
print(f"Average response time: {avg_rt:.2f}")
```

Output-



```
PS C:\Users\aryan> & C:/Users/aryan/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/
Process AT      BT      Priority        CT      TAT     WT      RT
p1      0       4       2               15      15      11      11
p2      1       3       3               12      11      8       8
p3      2       1       4               3       1       0       0
p4      3       5       5               10      7       2       2
p5      4       2       5               6       2       0       0

Average turnaround time: 7.20
Average waiting time: 4.20
Average response time: 4.20
PS C:\Users\aryan>
```