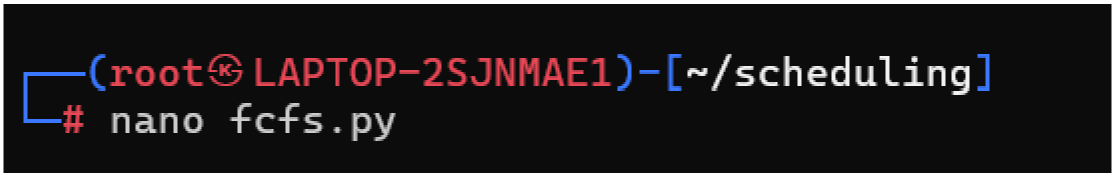
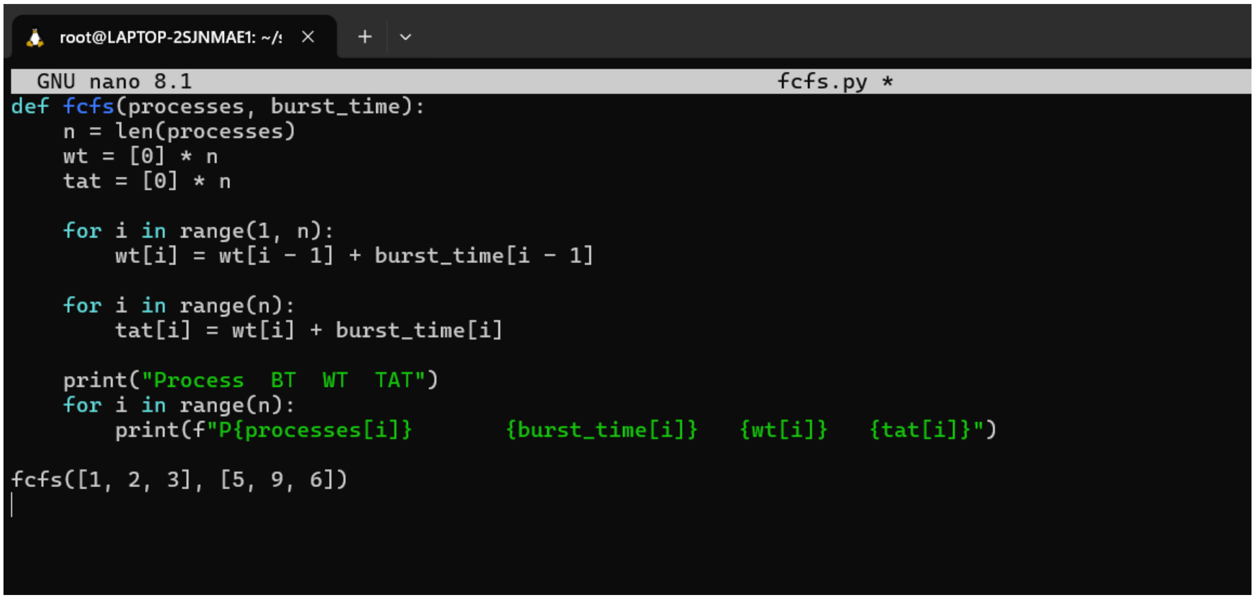
Scheduling Algorithms

1. First Come First Serve (FCFS)

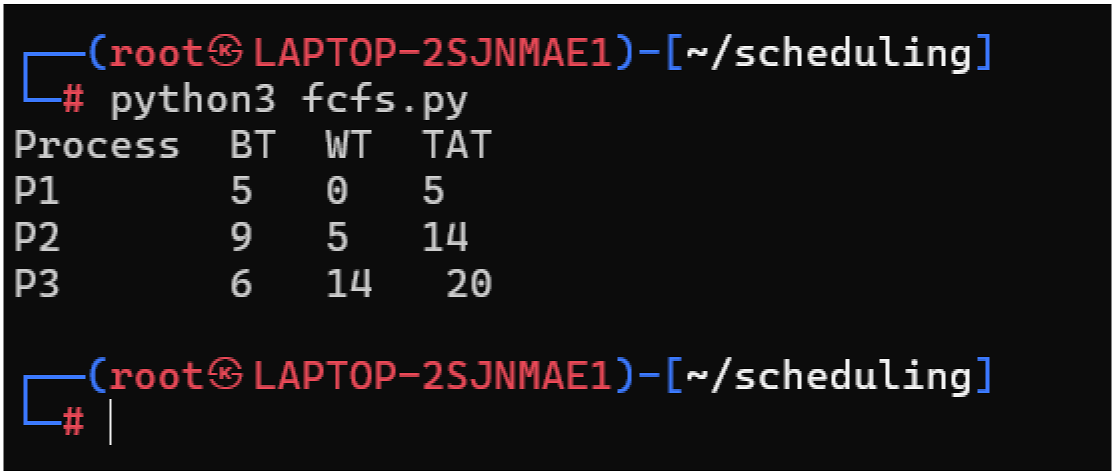
Sol.-



```
┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─# nano fcfs.py
```



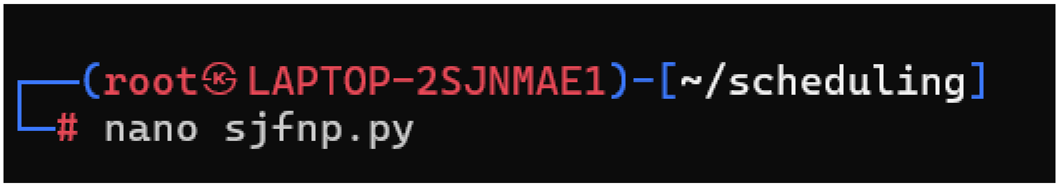
```
  GNU nano 8.1                                              fcfs.py *
def fcfs(processes, burst_time):
    n = len(processes)
    wt = [0] * n
    tat = [0] * n

    for i in range(1, n):
        wt[i] = wt[i - 1] + burst_time[i - 1]

    for i in range(n):
        tat[i] = wt[i] + burst_time[i]

    print("Process  BT  WT  TAT")
    for i in range(n):
        print(f"P{processes[i]}       {burst_time[i]}   {wt[i]}   {tat[i]}")

fcfs([1, 2, 3], [5, 9, 6])
```



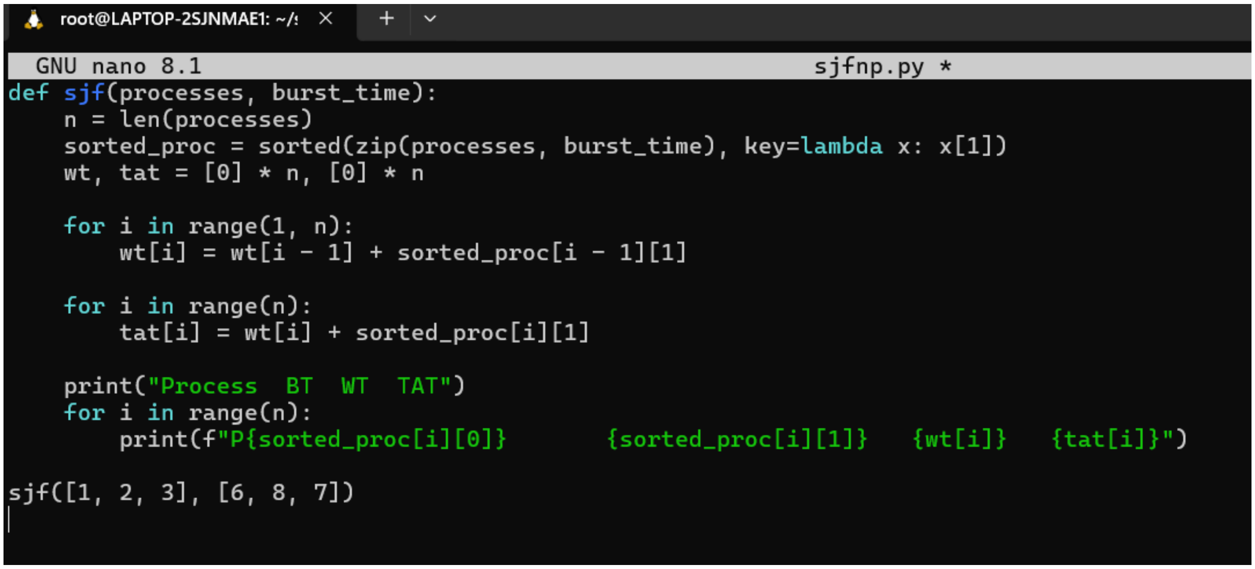
```
┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─# python3 fcfs.py
Process  BT  WT  TAT
P1       5   0   5
P2       9   5   14
P3       6   14   20

┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─#
```

2. (i) Shortest Job First (SJF) – Non-preemptive

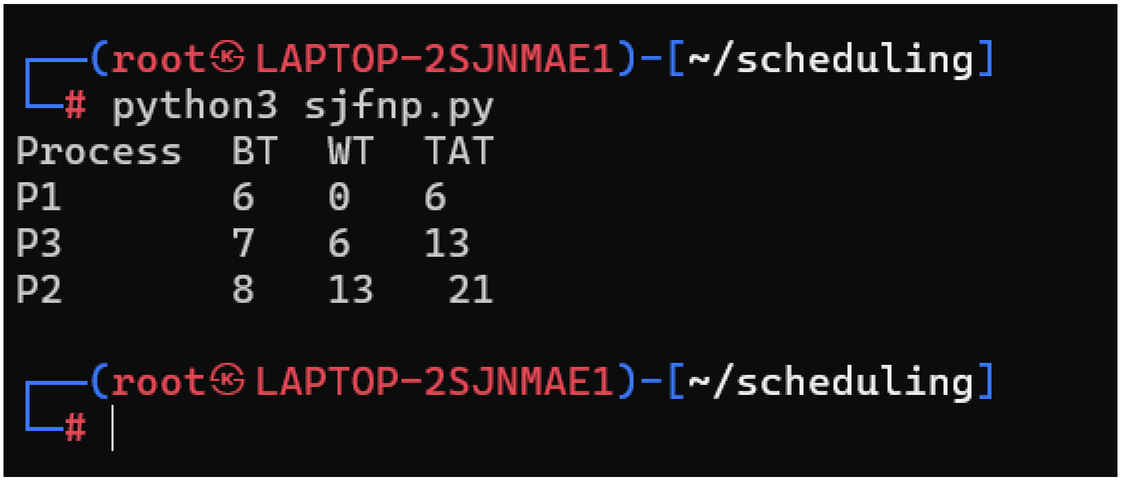
Sol. –



```
┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─# nano sjfnp.py
```



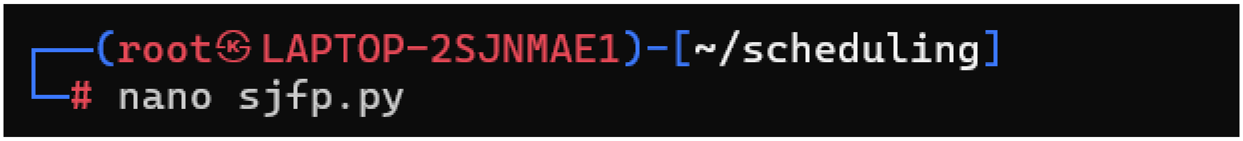
```
  GNU nano 8.1                                             sjfnp.py *
def sjf(processes, burst_time):
    n = len(processes)
    sorted_proc = sorted(zip(processes, burst_time), key=lambda x: x[1])
    wt, tat = [0] * n, [0] * n

    for i in range(1, n):
        wt[i] = wt[i - 1] + sorted_proc[i - 1][1]

    for i in range(n):
        tat[i] = wt[i] + sorted_proc[i][1]

    print("Process  BT  WT  TAT")
    for i in range(n):
        print(f"P{sorted_proc[i][0]}       {sorted_proc[i][1]}   {wt[i]}   {tat[i]}")

sjf([1, 2, 3], [6, 8, 7])
```



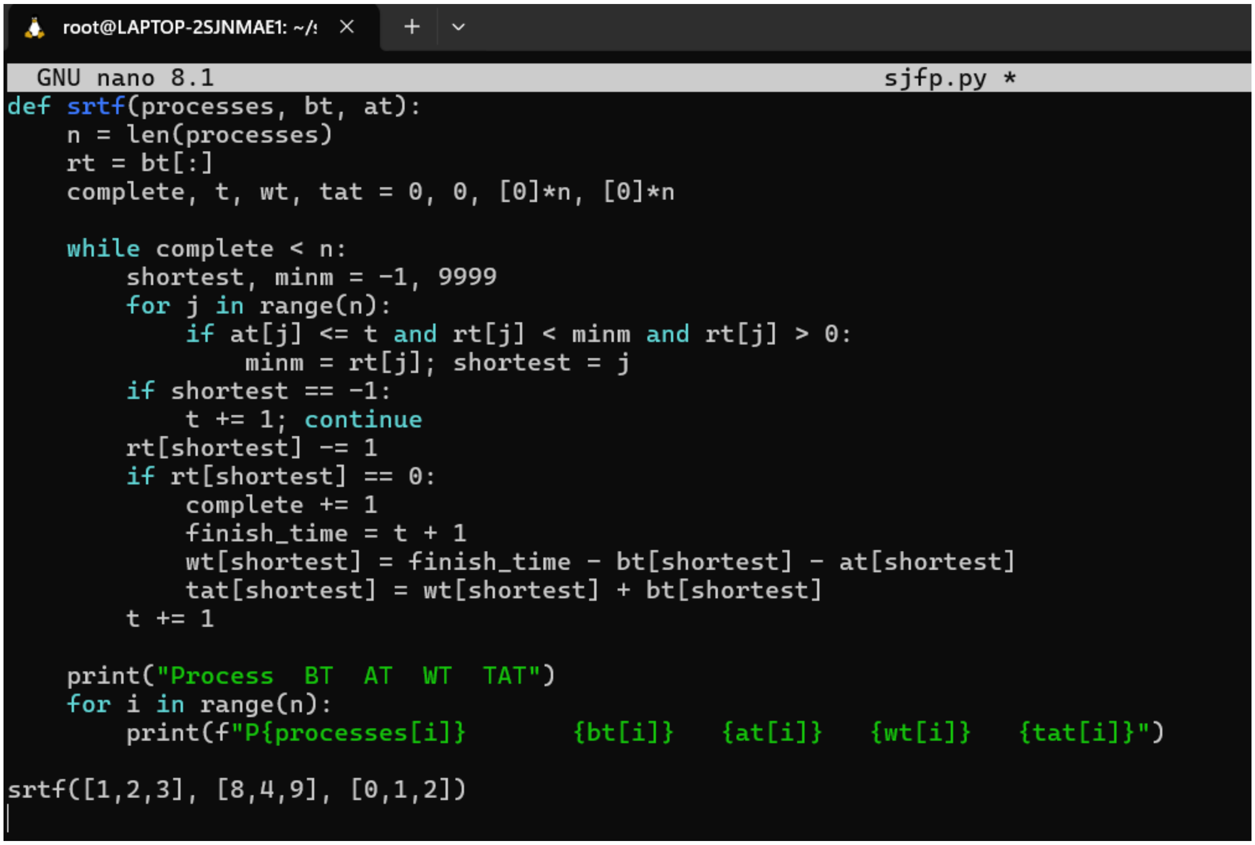
```
┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─# python3 sjfnp.py
Process  BT  WT  TAT
P1       6   0   6
P3       7   6   13
P2       8   13   21

┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─#
```

(ii) Shortest Remaining Time First (SRTF) – Preemptive SJF

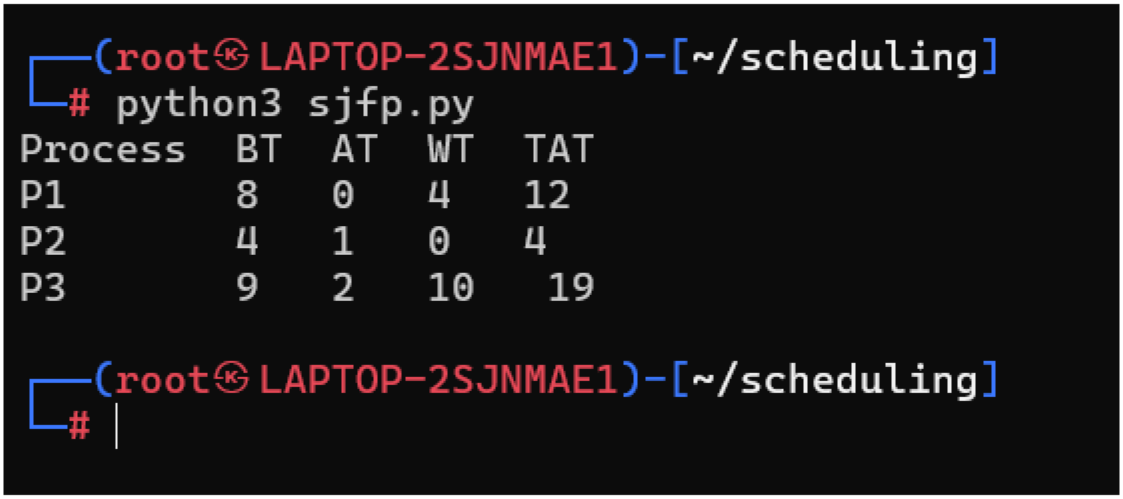
Sol.-



```
┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─# nano sjfp.py
```



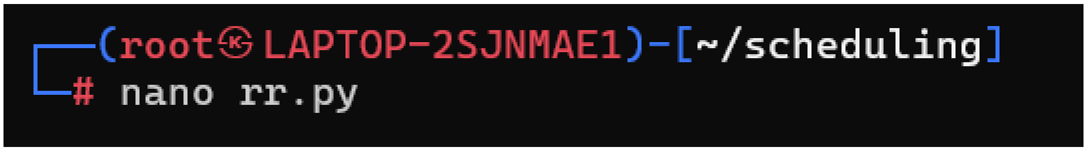
```
  GNU nano 8.1                                                    sjfp.py *
def srtf(processes, bt, at):
    n = len(processes)
    rt = bt[:]
    complete, t, wt, tat = 0, 0, [0]*n, [0]*n

    while complete < n:
        shortest, minm = -1, 9999
        for j in range(n):
            if at[j] <= t and rt[j] < minm and rt[j] > 0:
                minm = rt[j]; shortest = j
        if shortest == -1:
            t += 1; continue
        rt[shortest] -= 1
        if rt[shortest] == 0:
            complete += 1
            finish_time = t + 1
            wt[shortest] = finish_time - bt[shortest] - at[shortest]
            tat[shortest] = wt[shortest] + bt[shortest]
        t += 1

    print("Process  BT  AT  WT  TAT")
    for i in range(n):
        print(f"P{processes[i]}       {bt[i]}   {at[i]}   {wt[i]}   {tat[i]}")

srtf([1,2,3], [8,4,9], [0,1,2])
```



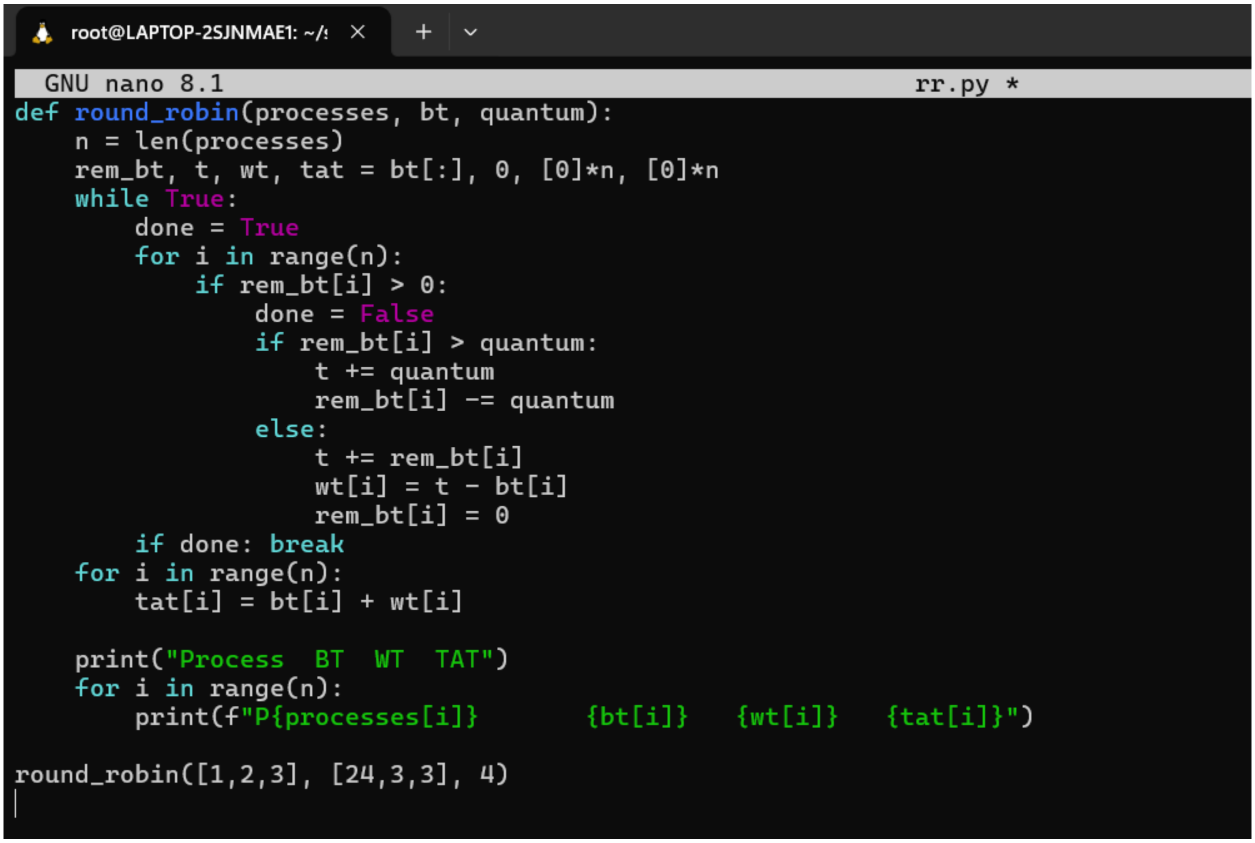
```
┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─# python3 sjfp.py
Process  BT  AT  WT  TAT
P1       8   0   4   12
P2       4   1   0   4
P3       9   2   10   19

┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─#
```

3. Round Robin (RR)

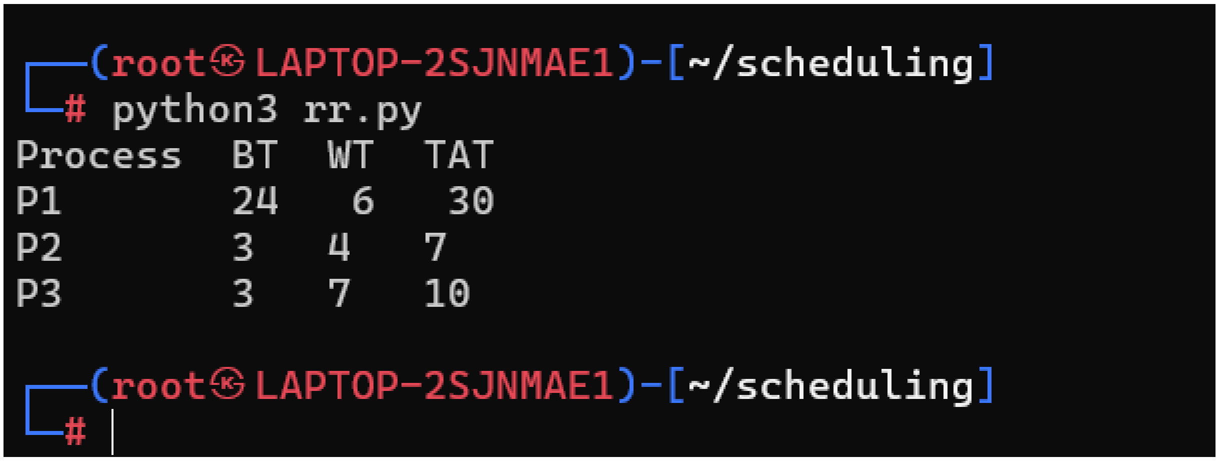
Sol.-



```
┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─# nano rr.py
```



```
  GNU nano 8.1                                                  rr.py *
def round_robin(processes, bt, quantum):
    n = len(processes)
    rem_bt, t, wt, tat = bt[:], 0, [0]*n, [0]*n
    while True:
        done = True
        for i in range(n):
            if rem_bt[i] > 0:
                done = False
                if rem_bt[i] > quantum:
                    t += quantum
                    rem_bt[i] -= quantum
                else:
                    t += rem_bt[i]
                    wt[i] = t - bt[i]
                    rem_bt[i] = 0
        if done: break
    for i in range(n):
        tat[i] = bt[i] + wt[i]

    print("Process  BT  WT  TAT")
    for i in range(n):
        print(f"P{processes[i]}       {bt[i]}   {wt[i]}   {tat[i]}")

round_robin([1,2,3], [24,3,3], 4)
```



```
┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─# python3 rr.py
Process  BT  WT  TAT
P1       24   6   30
P2       3   4   7
P3       3   7   10

┌──(root㉿LAPTOP-2SJNMAE1)-[~/scheduling]
└─#
```