**Experiment-6**

**(CPU Scheduling)**

**1) Shortest Remaining Time First (SRTF):**

**Code:**

n = int(input('Enter no of processes: '))

bt = [0] \* (n + 1)

at = [0] \* (n + 1)

abt = [0] \* (n + 1)

for i in range(n):

abt[i] = int(input('Enter the burst time for process {} : '.format(i + 1)))

at[i] = int(input('Enter the arrival time for process {} : '.format(i + 1)))

bt[i] = [abt[i], at[i], i]

bt.pop(-1)

print(abt)

print(bt)

sumbt = 0

i = 0

ll = []

for i in range(0, sum(abt)):

l = [j for j in bt if j[1] <= i]

l.sort(key=lambda x: x[0])

print(l, l[0][2])

bt[bt.index(l[0])][0] -= 1

for k in bt:

if k[0] == 0:

t = bt.pop(bt.index(k))

ll.append([k, i + 1])

print(ll)

ct = [0] \* (n + 1)

tat = [0] \* (n + 1)

wt = [0] \* (n + 1)

for i in ll:

print(i, i[0], i[1], i[0][2])

ct[i[0][2]] = i[1]

for i in range(len(ct)):

tat[i] = ct[i] - at[i]

wt[i] = tat[i] - abt[i]

ct.pop(-1)

wt.pop(-1)

tat.pop(-1)

abt.pop(-1)

at.pop(-1)

print('BT\tAT\tCT\tTAT\tWT')

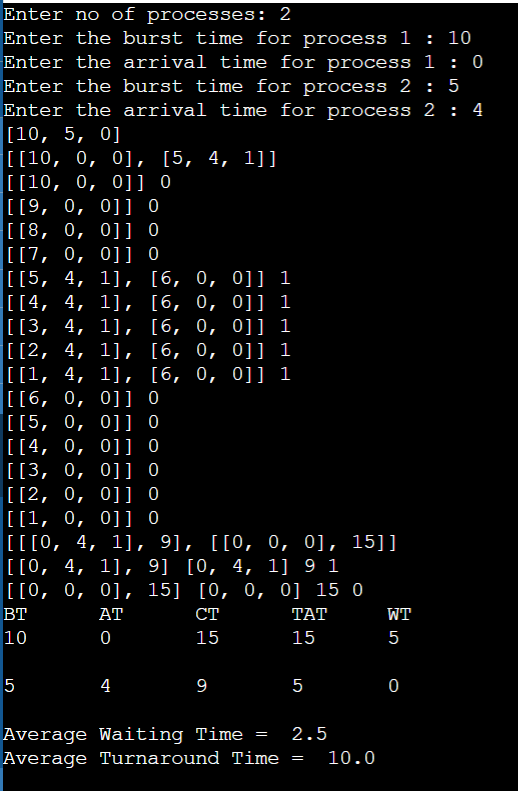
for i in range(len(ct)):

print("{}\t{}\t{}\t{}\t{}\n".format(abt[i], at[i], ct[i], tat[i], wt[i]))

print('Average Waiting Time = ', sum(wt)/len(wt))

print('Average Turnaround Time = ', sum(tat)/len(tat))

**Output:**



**2) Round Robin:**

**Code:**

if \_\_name\_\_ == '\_\_main\_\_':

print("Enter Total Process Number: ")

total\_p\_no = int(input())

total\_time = 0

total\_time\_counted = 0

proc = []

wait\_time = 0

turnaround\_time = 0

for \_ in range(total\_p\_no):

print("Enter process arrival time and burst time")

input\_info = list(map(int, input().split(" ")))

arrival, burst, remaining\_time = input\_info[0], input\_info[1], input\_info[1]

proc.append([arrival, burst, remaining\_time, 0])

total\_time += burst

print("Enter time quantum")

time\_quantum = int(input())

while total\_time != 0:

for i in range(len(proc)):

if proc[i][2] <= time\_quantum and proc[i][2] >= 0:

total\_time\_counted += proc[i][2]

total\_time -= proc[i][2]

proc[i][2] = 0

elif proc[i][2] > 0:

proc[i][2] -= time\_quantum

total\_time -= time\_quantum

total\_time\_counted += time\_quantum

if proc[i][2] == 0 and proc[i][3] != 1:

wait\_time += total\_time\_counted - proc[i][0] - proc[i][1]

turnaround\_time += total\_time\_counted - proc[i][0]

proc[i][3] = 1

print("\nAvg Waiting Time is ", (wait\_time \* 1) / total\_p\_no)

print("Avg Turnaround Time is ", (turnaround\_time \* 1) / total\_p\_no)

**Output:**

