

Write a program to simulate the following non-pre-emptive CPU scheduling algorithms to find turnaround time and waiting time.

a) FCFS

CODE:

```
n = int(input("Enter no. of processes: "))
process = list(map(str, input("Process: ").split()))
At = list(map(int, input("Arrival time: ").split()))
Bt = list(map(int, input("Burst time: ").split()))
gc = []
ct = [0]*n
wt = [0]*n
tat = [0]*n
rt = [0]*n
val = 0
for i in range(n):
    x = At.index(min(At))

    if At[x]>val:
        val = At[x]

    gc = gc + [(process[x])]
    ct[x] = val + Bt[x]
    tat[x] = ct[x]-At[x]
    wt[x] = tat[x]-Bt[x]
    rt[x] = val-At[x]
    val += Bt[x]
    At[x]=99999

print("Processes      :", process)
print("Completion time :",ct)
print("Turn Around time:",tat)
print("Waiting time     :",wt)
print("Response time    :",rt)
print("Gantt Chart       :",gc)
print("Avg Turn Around Time:", round(sum(tat)/n,3))
print("Avg Wating Time     :", round(sum(wt)/n,3))
```

INPUT & OUTPUT:

```

PS C:\Users\hp\Desktop\Langs\OS> python -u "c:\Users\hp\Desktop\Langs\OS\fcfs.py"
Enter no. of processes: 5
Process: p1 p2 p3 p4 p5
Arrival time: 2 0 2 3 4
Burst time: 2 1 3 5 4
Processes      : ['p1', 'p2', 'p3', 'p4', 'p5']
Completion time : [4, 1, 7, 12, 16]
Turn Around time: [2, 1, 5, 9, 12]
Waiting time    : [0, 0, 2, 4, 8]
Response time   : [0, 0, 2, 4, 8]
Gantt Chart     : ['p2', 'p1', 'p3', 'p4', 'p5']
Avg Turn Around Time: 5.8
Avg Wating Time  : 2.8

```

b) SJF

CODE:

```

n = int(input("Enter no. of processes: "))
process = list(map(str, input("Process: ").split()))
At = list(map(int, input("Arrival time:").split()))
Bt = list(map(int, input("Burst time: ").split()))
gc = []
ct = [0]*n
wt = [0]*n
tat = [0]*n
rt = [0]*n
val = 0
x = At.index(min(At))
for i in range(n):

    if At[x]>val:
        x = At.index(min(At))

    gc = gc + [(process[x])]
    ct[x] = val + Bt[x]
    tat[x] = ct[x]-At[x]
    wt[x] = tat[x]-Bt[x]
    rt[x] = val-At[x]
    val += Bt[x]
    Bt[x]=99999
    x = Bt.index(min(Bt))

print("Processes      :", process)
print("Completion time :",ct)
print("Turn Around time:",tat)
print("Waiting time     :",wt)
print("Response time    :",rt)
print("Gantt Chart      :",gc)
print("Avg Turn Around Time:", round(sum(tat)/n,3))
print("Avg Wating Time   :", round(sum(wt)/n,3))

```

INPUT & OUTPUT:

```
PS C:\Users\hp\Desktop\Langs\OS> python -u "c:\Users\hp\Desktop\Langs\OS\sjf.py"
Enter no. of processes: 5
Process: p1 p2 p3 p4 p5
Arrival time:2 1 4 0 2
Burst time: 1 5 1 6 3
Processes      : ['p1', 'p2', 'p3', 'p4', 'p5']
Completion time : [7, 16, 8, 6, 11]
Turn Around time: [5, 15, 4, 6, 9]
Waiting time    : [4, 10, 3, 0, 6]
Response time   : [4, 10, 3, 0, 6]
Gantt Chart     : ['p4', 'p1', 'p3', 'p5', 'p2']
Avg Turn Around Time: 7.8
Avg Wating Time : 4.6
```

c) Round Robin (pre-emptive)**CODE:**

```
n = int(input("Enter no. of processes: "))
process = list(map(str, input("Process: ").split()))
At = list(map(int, input("Arrival time: ").split()))
Bt = list(map(int, input("Burst time: ").split()))
t = int(input("Time Quantum: "))
At1 = sorted(At)
Bt1 = Bt.copy()
gc = []
rQ = []
ct = [0]*n
wt = [0]*n
tat = [0]*n
rt = [0]*n
val = cnt = flg = i = 0
s = sum(Bt)
while (max(ct)!=s):
    while(i<len(At1) and cnt>=At1[i]):
        rQ.append(At1[i])
        i+=1

    if flg==1:
        rQ.append(At[x])

    x = At.index(rQ[0])
    if process[x] not in gc:
        rt[x] = val-At[x]

    gc.append(process[x])
    rQ.remove(At[x])
```

```

if Bt[x]<=t and Bt[x]!=0:
    ct[x] = Bt[x] + cnt
    tat[x] = ct[x]-At[x]
    wt[x] = tat[x]-Bt1[x]
    val += Bt[x]
    cnt += Bt[x]
    Bt[x]=0
    flg=0

else:
    #flag to append incompletd processes
    Bt[x] = Bt[x]-t
    cnt+=t
    val = cnt
    flg=1

print("Processes      :", process)
print("Completion time :",ct)
print("Turn Around time:",tat)
print("Waiting time    :",wt)
print("Response time   :",rt)
print("Gantt Chart     :",gc)
print("Avg Turn Around Time:", round(sum(tat)/n,3))
print("Avg Wating Time   :", round(sum(wt)/n,3))

```

INPUT & OUTPUT:

```

PS C:\Users\hp\Desktop\Langs\OS> python -u "c:\Users\hp\Desktop\Langs\OS\rr.py"
Enter no. of processes: 5
Process: p1 p2 p3 p4 p5
Arrival time: 0 5 1 6 8
Burst time: 8 2 7 3 5
Time Quantum: 3
Processes      : ['p1', 'p2', 'p3', 'p4', 'p5']
Completion time : [22, 11, 23, 14, 25]
Turn Around time: [22, 6, 22, 8, 17]
Waiting time    : [14, 4, 15, 5, 12]
Response time   : [0, 4, 2, 5, 9]
Gantt Chart     : ['p1', 'p3', 'p1', 'p2', 'p4', 'p3', 'p5', 'p1', 'p3', 'p5']
Avg Turn Around Time: 15.0
Avg Wating Time   : 10.0

```

d) Priority

CODE:

```

n = int(input("Enter no. of processes: "))
process = list(map(str, input("Process: ").split()))
pri =list(map(int, input("Priority: ").split()))
At = list(map(int, input("Arrival time: ").split()))
Bt = list(map(int, input("Burst time: ").split()))

```

```
gc = []
ct = [0]*n
wt = [0]*n
tat = [0]*n
rt = [0]*n
val = 0
flag = 0
x = At.index(min(At))
for i in range(n):

    if At[x]>val:
        flag = 1
        x1 = x
        x2 = min(pri)
        pri[x1]=99999
        x = pri.index(min(pri))

    gc = gc + [(process[x])]
    ct[x] = val + Bt[x]
    tat[x] = ct[x]-At[x]
    wt[x] = tat[x]-Bt[x]
    rt[x] = val-At[x]
    val += Bt[x]
    pri[x]=99999
    At[x]=99999
    x = pri.index(min(pri))
    if flag==1:
        x=x1
        pri[x1]=x2
        flag = 0

print("Processes      :", process)
print("Completion time :",ct)
print("Turn Around time:",tat)
print("Waiting time    :",wt)
print("Response time   :",rt)
print("Gantt Chart     :",gc)
print("Avg Turn Around Time:", round(sum(tat)/n,2))
print("Avg Wating Time   :", round(sum(wt)/n,2))
```

INPUT & OUTPUT:

```
PS C:\Users\hp\Desktop\Langs\OS> python -u "c:\Users\hp\Desktop\Langs\OS\priority.py"
Enter no. of processes: 7
Process:p1 p2 p3 p4 p5 p6 p7
Priority: 3 4 4 5 2 6 1
Arrival time: 0 1 3 4 5 6 10
Burst time: 8 2 4 1 6 5 1
Processes      : ['p1', 'p2', 'p3', 'p4', 'p5', 'p6', 'p7']
Completion time : [8, 17, 21, 22, 14, 27, 15]
Turn Around time: [8, 16, 18, 18, 9, 21, 5]
Waiting time    : [0, 14, 14, 17, 3, 16, 4]
Response time   : [0, 14, 14, 17, 3, 16, 4]
Gantt Chart     : ['p1', 'p5', 'p7', 'p2', 'p3', 'p4', 'p6']
Avg Turn Around Time: 13.57
Avg Wating Time   : 9.71
PS C:\Users\hp\Desktop\Langs\OS>
```