

ROUND ROBIN**OUTPUT:**

```

Enter number of processes: 3
Enter Process 1 ID (char): A
Enter Arrival Time: 0
Enter Burst Time: 4
Enter Process 2 ID (char): B
Enter Arrival Time: 2
Enter Burst Time: 2
Enter Process 3 ID (char): C
Enter Arrival Time: 3
Enter Burst Time: 2
Enter Time Quantum: 2

ID      AT      BT      CT      TT      WT
A       0       4       6       6       2
B       2       2       4       2       0
C       3       2       8       5       3

Avg WT = 1.66667
Avg TT = 4.33333

Gantt Chart:
| A | B | A | C |
0  2  4  6  8

```

EXPLANATION/COMPUTATION

TIME	EXPLANATION
0	ARRIVES A
2	A GO TO READY QUEUE BECAUSE B WILL RUN. RQ[A]
4	B FINISHED RUN A. QR: [NULL]
6	A FINISHED RUN C
C	FINISHED

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n, i, j, time = 0, done = 0, quantum;
6      char id[20];
7      int at[20];
8      int bt[20];
9      int rt[20];
10     int ct[20], tt[20], wt[20];
11     int gantt[200], gTime[201], gLen = 0;
12     int queue[100];
13     int front = 0, rear = 0;
14     bool inQueue[20] = {false};
15     float totWT = 0, totTT = 0;
16
17     cout << "Enter number of processes: ";
18     cin >> n;
19     for (i = 0; i < n; i++)
20     {
21         cout << "Enter Process " << i + 1 << " ID (char): ";
22         cin >> id[i];
23         cout << "Enter Arrival Time: ";
24         cin >> at[i];
25         cout << "Enter Burst Time: ";
26         cin >> bt[i];
27         rt[i] = bt[i];
28     }
29     cout << "Enter Time Quantum: ";
30     cin >> quantum;
31
32     queue[rear++] = 0;
33     inQueue[0] = true;
34     time = at[0];
35     while (done < n)
36     {
37         if (front == rear)
38         {
39             for (i = 0; i < n; i++)
40             {
41                 if (!inQueue[i] && rt[i] > 0)
42                 {
43                     queue[rear++] = i;
44                     inQueue[i] = true;
45                     time = at[i];
46                     break;
47                 }
48             }
49             if (front == rear)
50             {
51                 time++;
52                 continue;
53             }
54         }
55         int idx = queue[front++];
56
57         int execTime = 0;
58         if (rt[idx] > quantum)
59         {
60             execTime = quantum;
61         }
62         else
63         {
64             execTime = rt[idx];
65         }
66
67         if (gLen == 0 || gantt[gLen - 1] != idx)
68         {
69             gTime[gLen] = time;
70

```

```

71             int idx = queue[front++];
72
73             int execTime = 0;
74             if (rt[idx] > quantum)
75             {
76                 execTime = quantum;
77             }
78             else
79             {
80                 execTime = rt[idx];
81             }
82
83             if (gLen == 0 || gantt[gLen - 1] != idx)
84             {
85                 gTime[gLen] = time;
86                 gantt[gLen++] = idx;
87             }
88
89             rt[idx] -= execTime;
90             time += execTime;
91
92             for (i = 0; i < n; i++)
93             {
94                 if (!inQueue[i] && at[i] <= time && rt[i] > 0)
95                 {
96                     queue[rear++] = i;
97                     inQueue[i] = true;
98                 }
99             }
100             if (rt[idx] == 0)
101             {
102                 ct[idx] = time;
103                 tt[idx] = ct[idx] - at[idx];
104                 wt[idx] = tt[idx] - bt[idx];
105                 totWT += wt[idx];
106                 totTT += tt[idx];
107                 done++;
108             }
109             else
110             {
111                 queue[rear++] = idx;
112             }
113         }
114         gTime[gLen] = time;
115
116         cout << "\nID\tAT\tBT\tCT\tTT\tWT\n";
117         for (i = 0; i < n; i++)
118         {
119             cout << id[i] << "\t" << at[i] << "\t" << bt[i] << "\t"
120                 << ct[i] << "\t" << tt[i] << "\t" << wt[i] << "\n";
121         }
122         cout << "\nAvg WT = " << totWT / n;
123         cout << "\nAvg TT = " << totTT / n;
124
125         cout << "\n\nGantt Chart:\n ";
126         for (i = 0; i < gLen; i++)
127         {
128             cout << id[gantt[i]] << " | ";
129         }
130         cout << "\n";
131         for (i = 0; i <= gLen; i++)
132         {
133             cout << gTime[i] << " ";
134         }
135         cout << "\n";
136         return 0;
137     }
138 }

```

