Họ và tên: Lê Như Thực

Mã số sinh viên: 24521747

Lớp: IT007.Q112.1

HỆ ĐIỀU HÀNH

BÁO CÁO LAB 4

CHECKLIST

3.5. BÀI TẬP THỰC HÀNH

	BT 1	BT 2
Vẽ lưu đồ giải thuật	☒	☒
Chạy tay lưu đồ giải thuật	☒	☒
Hiện thực code	☒	☒
Chạy code và kiểm chứng	☒	☒

3.6. BÀI TẬP ÔN TẬP

	BT 1
Vẽ lưu đồ giải thuật	☒
Chạy tay lưu đồ giải thuật	☒
Hiện thực code	☒
Chạy code và kiểm chứng	

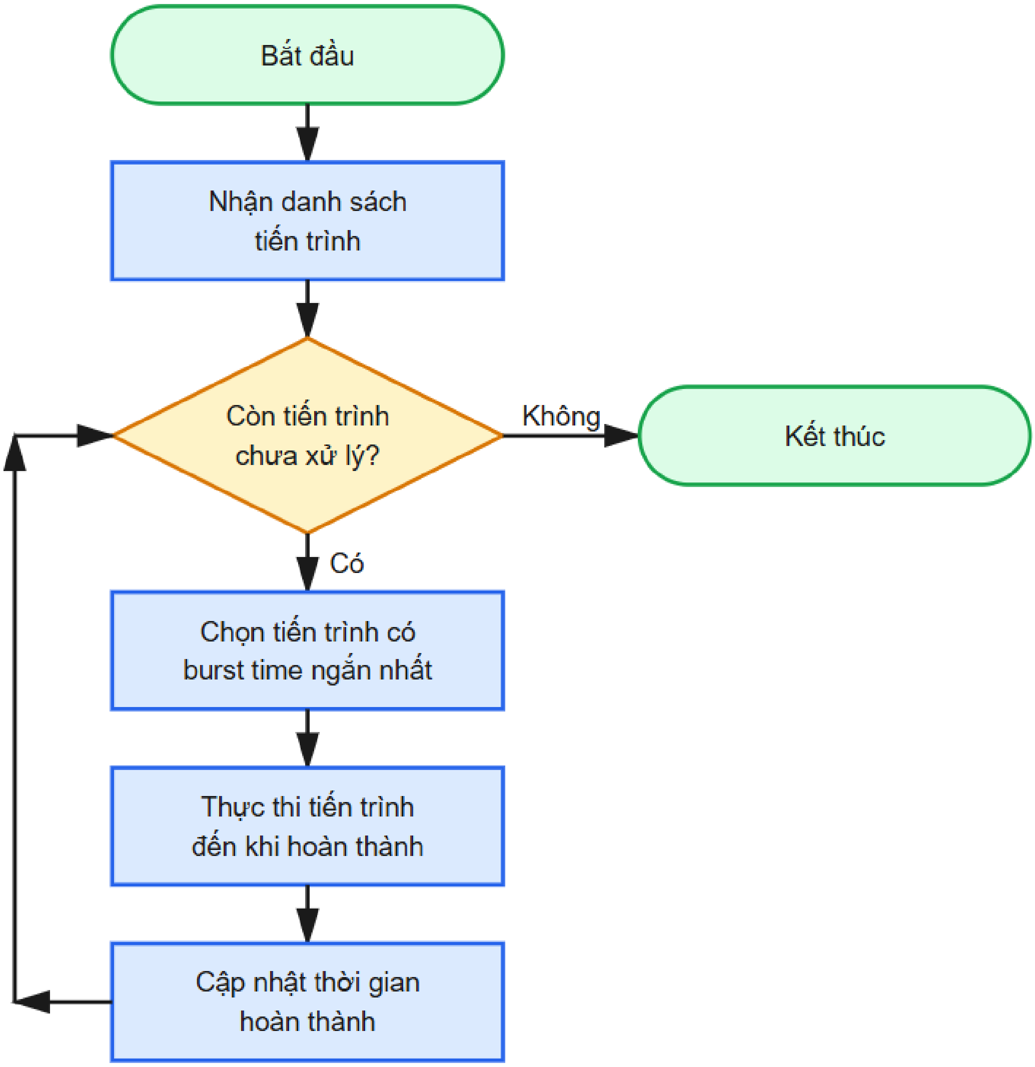
Tự chấm điểm: 10

**Lưu ý: Xuất báo cáo theo định dạng PDF, đặt tên theo cú pháp:*

<Tên nhóm>_LAB3.pdf

2.5. BÀI TẬP THỰC HÀNH

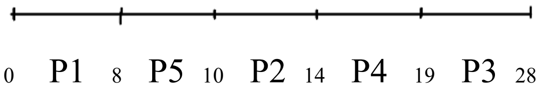
1. Giải thuật Shortest-Job-First


Bắt đầu
Nhận danh sách tiến trình
Còn tiến trình chưa xử lý?
Không
Kết thúc
Có
Chọn tiến trình có burst time ngắn nhất
Thực thi tiến trình đến khi hoàn thành
Cập nhật thời gian hoàn thành

Hình 1 : lưu đồ giải thuật SJF

PID	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

P5	4	2

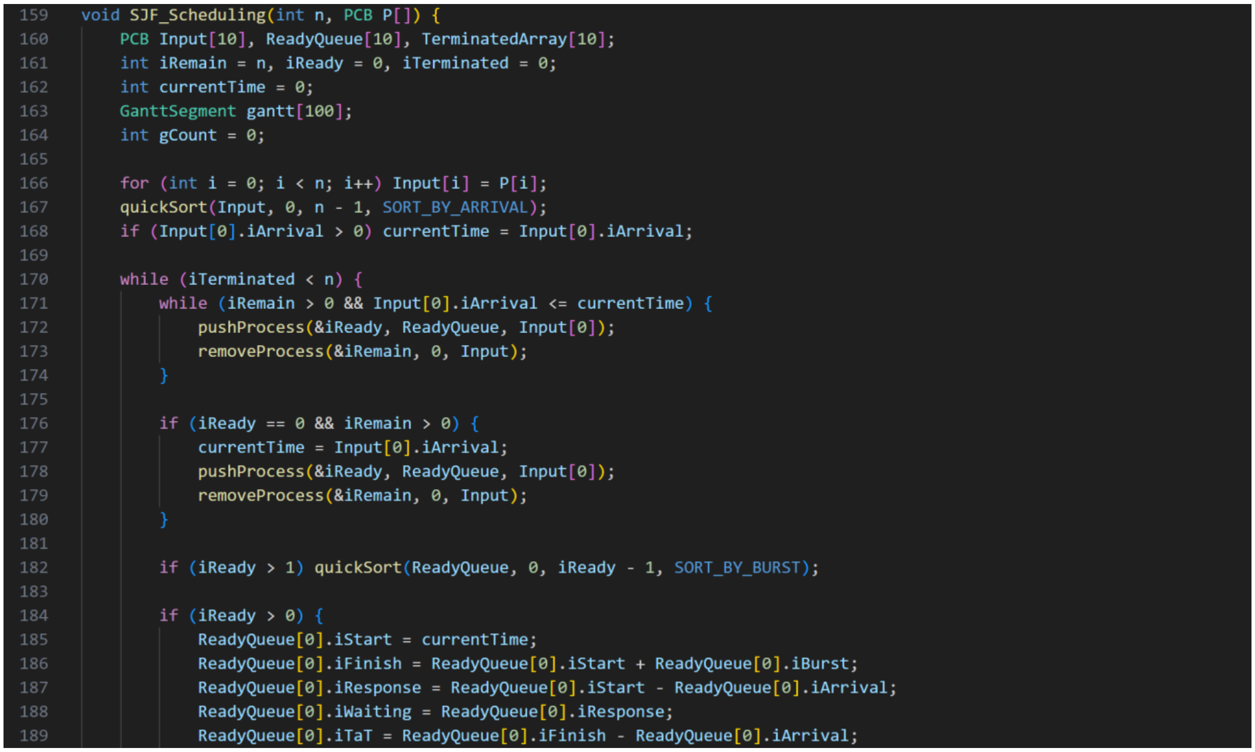


PID	Arrival	Burst	Start	Finish	Waiting	Response	TaT
P1	0	8	0	8	0	0	8
P2	1	4	10	14	9	9	13
P3	2	9	19	28	17	17	26
P4	3	5	14	19	11	11	16
P5	4	2	8	10	4	4	6

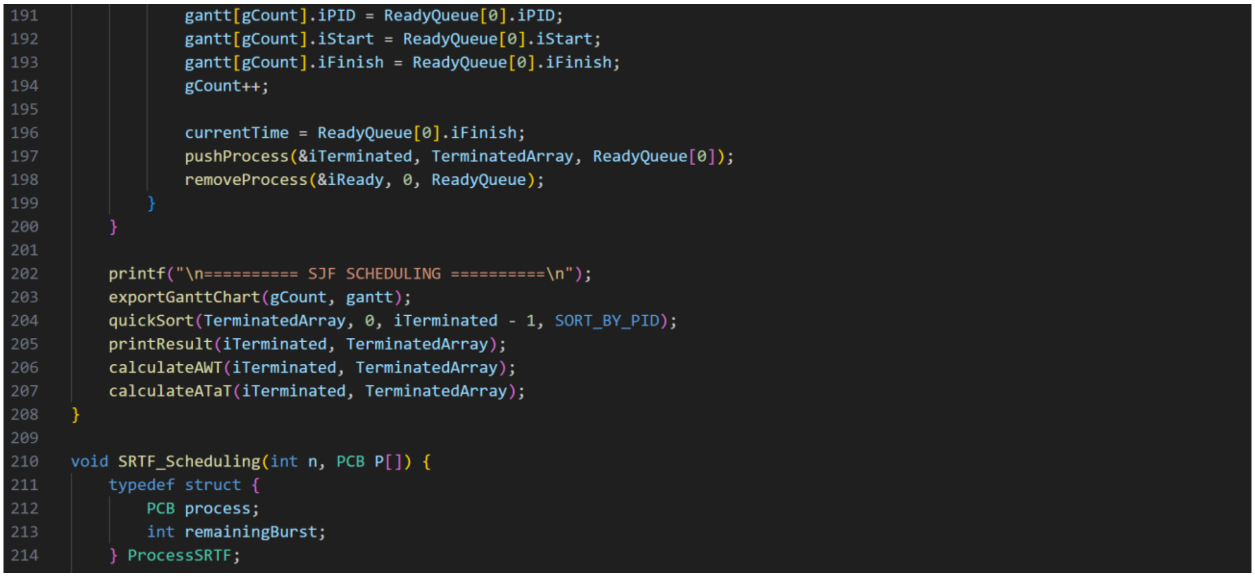
Waiting Time: (0+9+17+11+4)/5 = 8.2

Turnaround Time: (8+13+26+16+6)/5 = 13.8

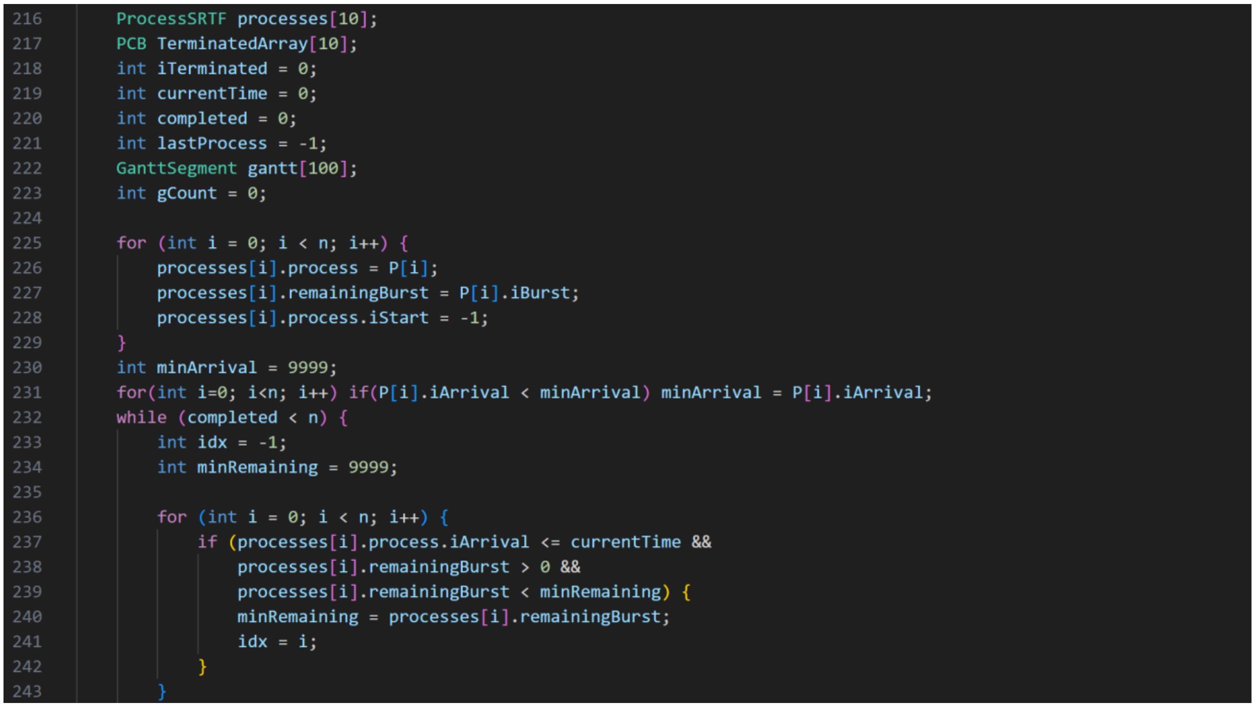
Response Time: (0+9+17+11+4)/5 = 8.2



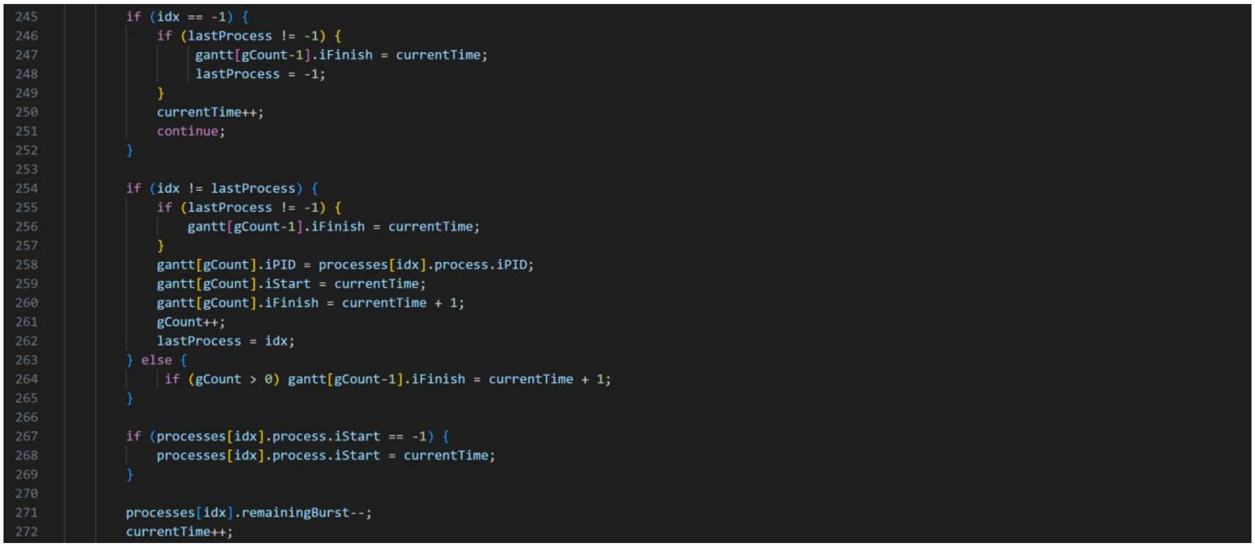
```
void SJF_Scheduling(int n, PCB P[]) {
    PCB Input[10], ReadyQueue[10], TerminatedArray[10];
    int iRemain = n, iReady = 0, iTerminated = 0;
    int currentTime = 0;
    GanttSegment gantt[100];
    int gCount = 0;

    for (int i = 0; i < n; i++) Input[i] = P[i];
    quickSort(Input, 0, n - 1, SORT_BY_ARRIVAL);
    if (Input[0].iArrival > 0) currentTime = Input[0].iArrival;

    while (iTerminated < n) {
        while (iRemain > 0 && Input[0].iArrival <= currentTime) {
            pushProcess(&iReady, ReadyQueue, Input[0]);
            removeProcess(&iRemain, 0, Input);
        }

        if (iReady == 0 && iRemain > 0) {
            currentTime = Input[0].iArrival;
            pushProcess(&iReady, ReadyQueue, Input[0]);
            removeProcess(&iRemain, 0, Input);
        }

        if (iReady > 1) quickSort(ReadyQueue, 0, iReady - 1, SORT_BY_BURST);

        if (iReady > 0) {
            ReadyQueue[0].iStart = currentTime;
            ReadyQueue[0].iFinish = ReadyQueue[0].iStart + ReadyQueue[0].iBurst;
            ReadyQueue[0].iResponse = ReadyQueue[0].iStart - ReadyQueue[0].iArrival;
            ReadyQueue[0].iWaiting = ReadyQueue[0].iResponse;
            ReadyQueue[0].iTaT = ReadyQueue[0].iFinish - ReadyQueue[0].iArrival;
```



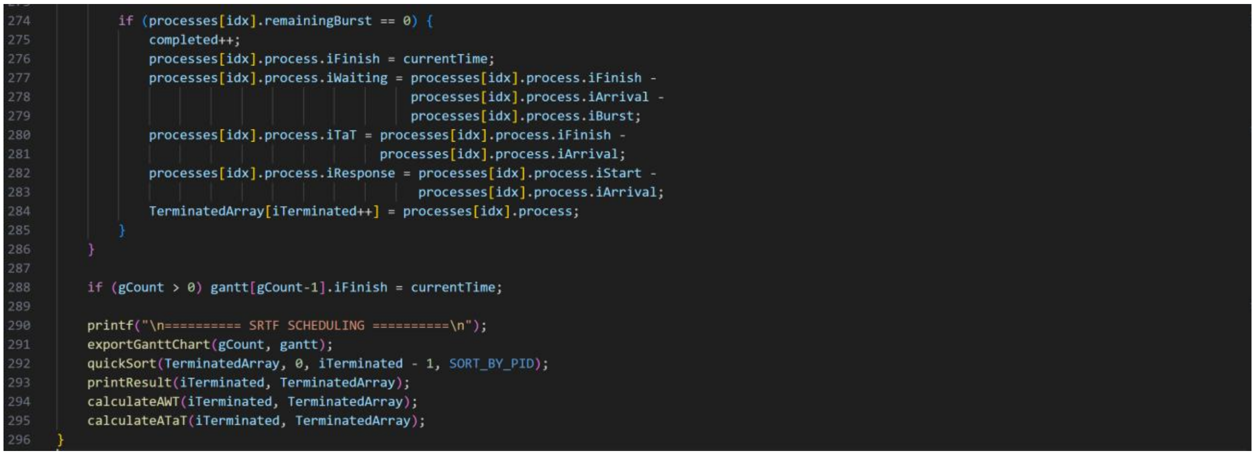
```
            gantt[gCount].iPID = ReadyQueue[0].iPID;
            gantt[gCount].iStart = ReadyQueue[0].iStart;
            gantt[gCount].iFinish = ReadyQueue[0].iFinish;
            gCount++;

            currentTime = ReadyQueue[0].iFinish;
            pushProcess(&iTerminated, TerminatedArray, ReadyQueue[0]);
            removeProcess(&iReady, 0, ReadyQueue);
        }
    }

    printf("\n========= SJF SCHEDULING =========\n");
    exportGanttChart(gCount, gantt);
    quickSort(TerminatedArray, 0, iTerminated - 1, SORT_BY_PID);
    printResult(iTerminated, TerminatedArray);
    calculateAWT(iTerminated, TerminatedArray);
    calculateATaT(iTerminated, TerminatedArray);
}

void SRTF_Scheduling(int n, PCB P[]) {
    typedef struct {
        PCB process;
        int remainingBurst;
    } ProcessSRTF;
```



```
    ProcessSRTF processes[10];
    PCB TerminatedArray[10];
    int iTerminated = 0;
    int currentTime = 0;
    int completed = 0;
    int lastProcess = -1;
    GanttSegment gantt[100];
    int gCount = 0;

    for (int i = 0; i < n; i++) {
        processes[i].process = P[i];
        processes[i].remainingBurst = P[i].iBurst;
        processes[i].process.iStart = -1;
    }
    int minArrival = 9999;
    for(int i=0; i<n; i++) if(P[i].iArrival < minArrival) minArrival = P[i].iArrival;
    while (completed < n) {
        int idx = -1;
        int minRemaining = 9999;

        for (int i = 0; i < n; i++) {
            if (processes[i].process.iArrival <= currentTime &&
                processes[i].remainingBurst > 0 &&
                processes[i].remainingBurst < minRemaining) {
                minRemaining = processes[i].remainingBurst;
                idx = i;
            }
        }
```



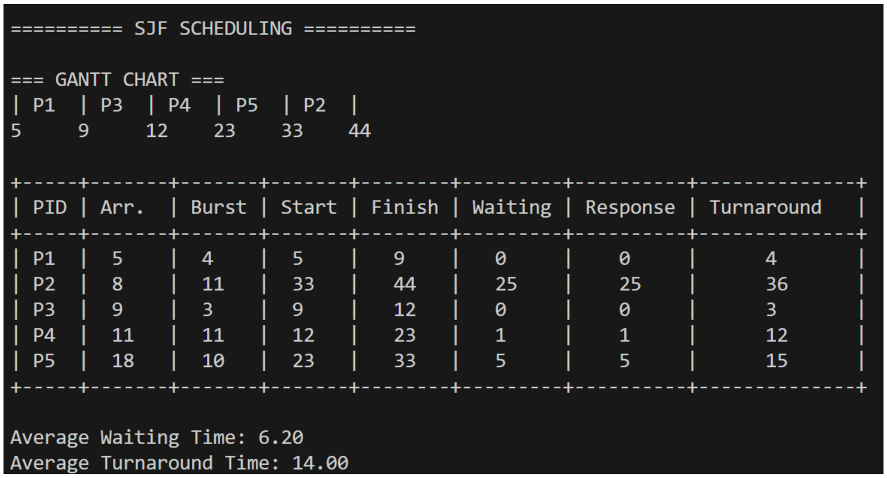
```
        if (idx == -1) {
            if (lastProcess != -1) {
                gantt[gCount-1].iFinish = currentTime;
                lastProcess = -1;
            }
            currentTime++;
            continue;
        }

        if (idx != lastProcess) {
            if (lastProcess != -1) {
                gantt[gCount-1].iFinish = currentTime;
            }
            gantt[gCount].iPID = processes[idx].process.iPID;
            gantt[gCount].iStart = currentTime;
            gantt[gCount].iFinish = currentTime + 1;
            gCount++;
            lastProcess = idx;
        } else {
            if (gCount > 0) gantt[gCount-1].iFinish = currentTime + 1;
        }

        if (processes[idx].process.iStart == -1) {
            processes[idx].process.iStart = currentTime;
        }

        processes[idx].remainingBurst--;
        currentTime++;
```



```
        if (processes[idx].remainingBurst == 0) {
            completed++;
            processes[idx].process.iFinish = currentTime;
            processes[idx].process.iWaiting = processes[idx].process.iFinish -
                                              processes[idx].process.iArrival -
                                              processes[idx].process.iBurst;
            processes[idx].process.iTaT = processes[idx].process.iFinish -
                                          processes[idx].process.iArrival;
            processes[idx].process.iResponse = processes[idx].process.iStart -
                                               processes[idx].process.iArrival;
            TerminatedArray[iTerminated++] = processes[idx].process;
        }
    }

    if (gCount > 0) gantt[gCount-1].iFinish = currentTime;

    printf("\n========== SRTF SCHEDULING ==========\n");
    exportGanttChart(gCount, gantt);
    quickSort(TerminatedArray, 0, iTerminated - 1, SORT_BY_PID);
    printResult(iTerminated, TerminatedArray);
    calculateAWT(iTerminated, TerminatedArray);
    calculateATaT(iTerminated, TerminatedArray);
}
```

Hình ảnh code

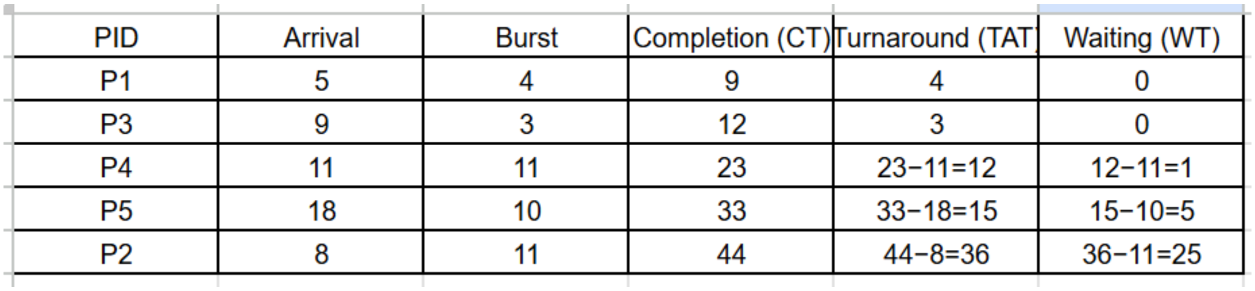
---Test 1---



```
========== SJF SCHEDULING ==========

=== GANTT CHART ===
| P1   | P3   | P4   | P5   | P2   |
5     9      12     23     33     44

+-----+-------+-------+-------+--------+---------+----------+-------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround  |
+-----+-------+-------+-------+--------+---------+----------+-------------+
| P1  |  5    |  4    |  5    |   9    |   0     |    0     |      4      |
| P2  |  8    |  11   |  33   |   44   |   25    |    25    |      36     |
| P3  |  9    |  3    |  9    |   12   |   0     |    0     |      3      |
| P4  |  11   |  11   |  12   |   23   |   1     |    1     |      12     |
| P5  |  18   |  10   |  23   |   33   |   5     |    5     |      15     |
+-----+-------+-------+-------+--------+---------+----------+-------------+

Average Waiting Time: 6.20
Average Turnaround Time: 14.00
```

| P1 | P3 | P4 | P5 | P2 |

5   9   12   23   33   44

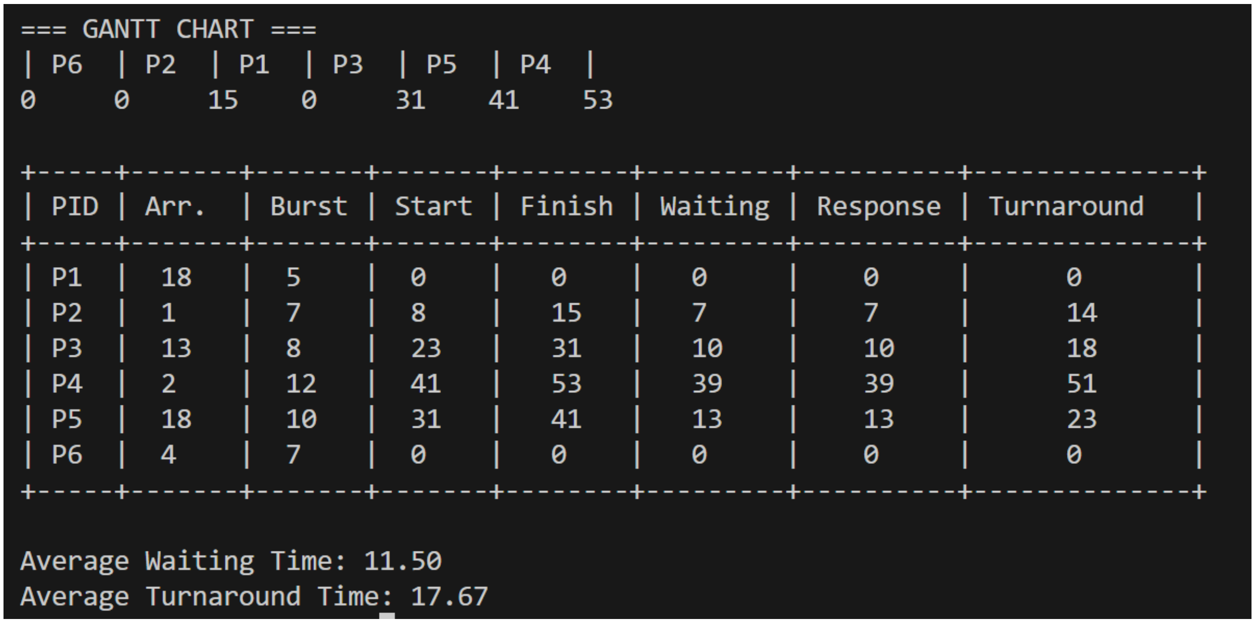


PID	Arrival	Burst	Completion (CT)	Turnaround (TAT	Waiting (WT)
P1	5	4	9	4	0
P3	9	3	12	3	0
P4	11	11	23	23−11=12	12−11=1
P5	18	10	33	33−18=15	15−10=5
P2	8	11	44	44−8=36	36−11=25

Thời gian chờ trung bình (Average Waiting Time): 6.2

Thời gian lưu lại trung bình (Average Turnaround Time): 14.0

---Test 2---



```
=== GANTT CHART ===
| P6  | P2  | P1  | P3  | P5  | P4  |
0     0     15    0     31    41    53

+-----+-------+-------+-------+--------+---------+----------+-------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround  |
+-----+-------+-------+-------+--------+---------+----------+-------------+
| P1  | 18    | 5     | 0     | 0      | 0       | 0        | 0           |
| P2  | 1     | 7     | 8     | 15     | 7       | 7        | 14          |
| P3  | 13    | 8     | 23    | 31     | 10      | 10       | 18          |
| P4  | 2     | 12    | 41    | 53     | 39      | 39       | 51          |
| P5  | 18    | 10    | 31    | 41     | 13      | 13       | 23          |
| P6  | 4     | 7     | 0     | 0      | 0       | 0        | 0           |
+-----+-------+-------+-------+--------+---------+----------+-------------+

Average Waiting Time: 11.50
Average Turnaround Time: 17.67
```

| P2 | P6 | P3 | P1 | P5 | P4 |

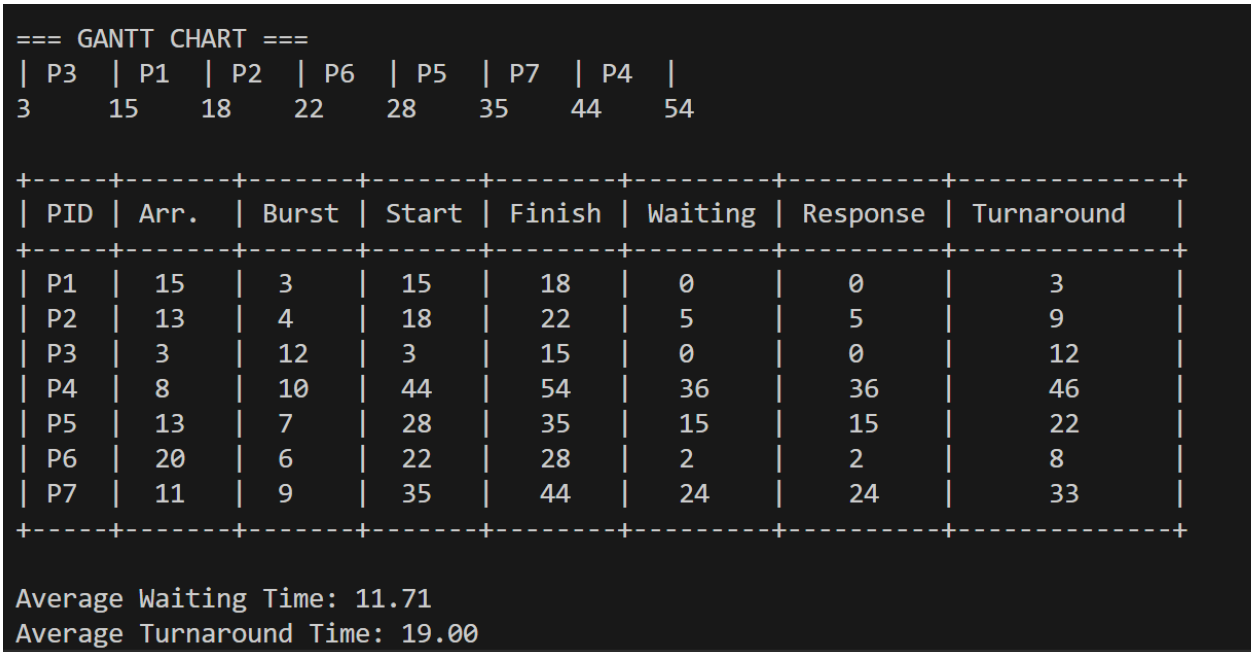
1 8 15 23 28 38 50

PID	Arrival	Burst	Completion (CT)	Turnaround (TAT)	Waiting (WT)
P1	18	5	28	28−18=10	10−5=5
P2	1	7	8	8−1=7	7−7=0
P3	13	8	23	23−13=10	10−8=2
P4	2	12	50	50−2=48	48−12=36
P5	18	10	38	38−18=20	20−10=10
P6	4	7	15	15−4=11	11−7=4

Thời gian chờ trung bình (Avg Waiting Time): 9.5

Thời gian lưu lại trung bình (Avg Turnaround Time): 17.67

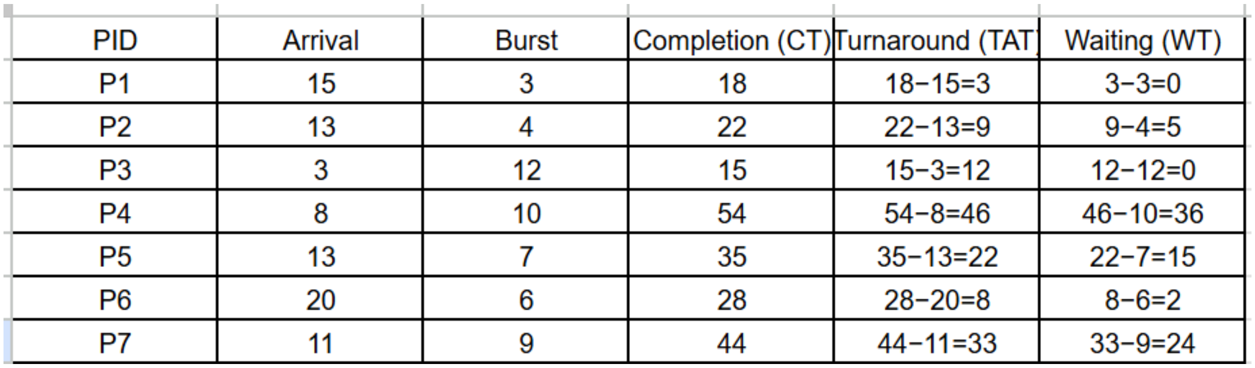
---Test 3---



```
=== GANTT CHART ===
| P3  | P1  | P2  | P6  | P5  | P7  | P4  |
3     15    18    22    28    35    44    54

+-----+-------+-------+-------+--------+---------+----------+-------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround  |
+-----+-------+-------+-------+--------+---------+----------+-------------+
| P1  |  15   | 3     |  15   |   18   |  0      |   0      |    3        |
| P2  |  13   | 4     |  18   |   22   |  5      |   5      |    9        |
| P3  |  3    | 12    |  3    |   15   |  0      |   0      |    12       |
| P4  |  8    | 10    |  44   |   54   |  36     |   36     |    46       |
| P5  |  13   | 7     |  28   |   35   |  15     |   15     |    22       |
| P6  |  20   | 6     |  22   |   28   |  2      |   2      |    8        |
| P7  |  11   | 9     |  35   |   44   |  24     |   24     |    33       |
+-----+-------+-------+-------+--------+---------+----------+-------------+

Average Waiting Time: 11.71
Average Turnaround Time: 19.00
```

| P3 | P1 | P2 | P6 | P5 | P7 | P4 |

3 15 18 22 28 35 44 54



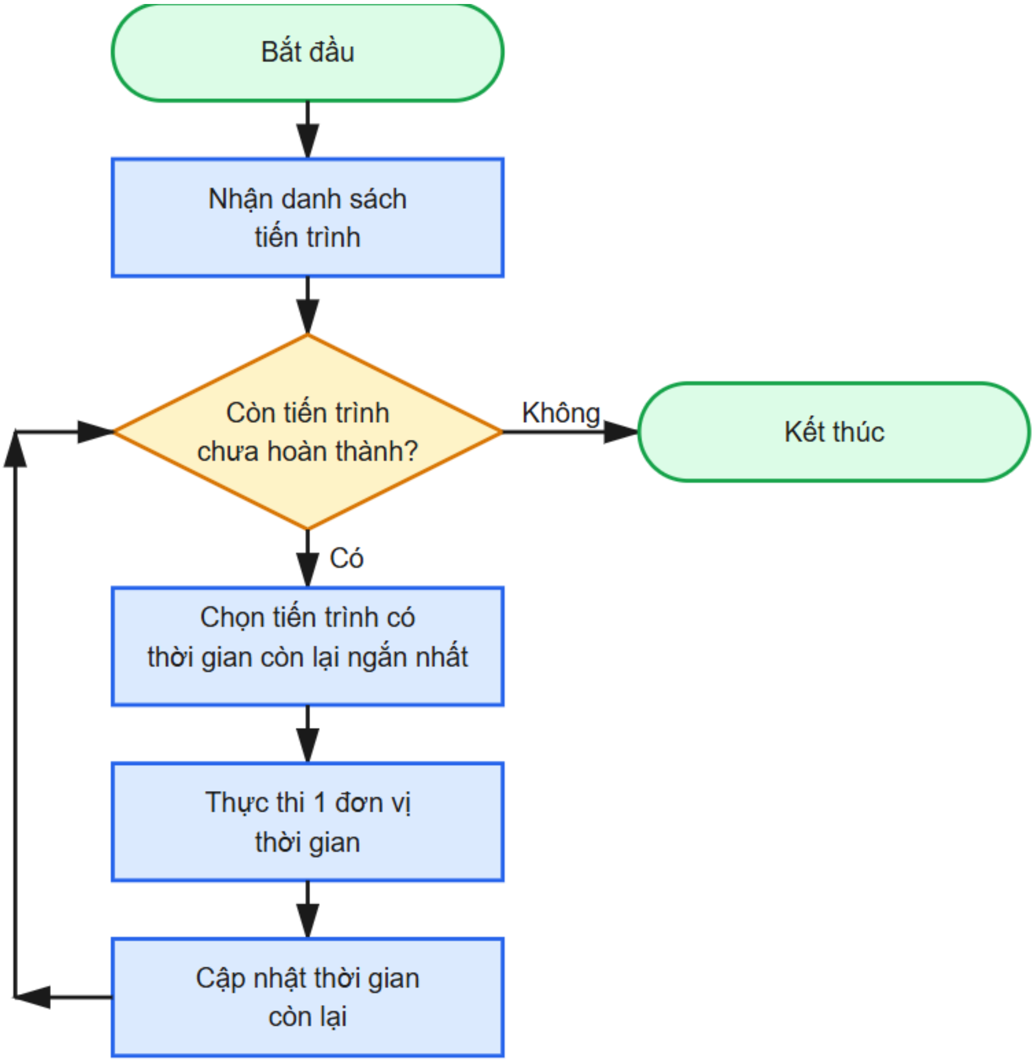
PID	Arrival	Burst	Completion (CT)	Turnaround (TAT)	Waiting (WT)
P1	15	3	18	18−15=3	3−3=0
P2	13	4	22	22−13=9	9−4=5
P3	3	12	15	15−3=12	12−12=0
P4	8	10	54	54−8=46	46−10=36
P5	13	7	35	35−13=22	22−7=15
P6	20	6	28	28−20=8	8−6=2
P7	11	9	44	44−11=33	33−9=24

Thời gian chờ trung bình (Average Waiting Time): 11.71

Thời gian lưu lại trung bình (Average Turnaround Time): 19.00

...

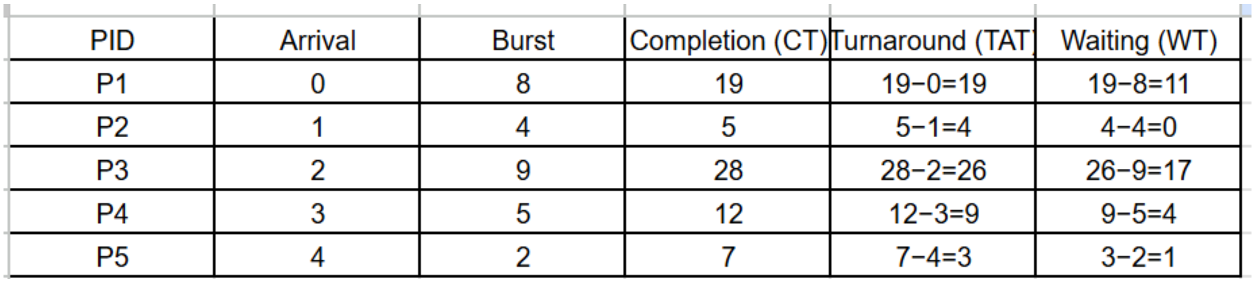
2. Giải thuật Shortest-Remaining-Time-First hoặc Round Robin


Bắt đầu
Nhận danh sách tiến trình
Còn tiến trình chưa hoàn thành?
Không
Kết thúc
Có
Chọn tiến trình có thời gian còn lại ngắn nhất
Thực thi 1 đơn vị thời gian
Cập nhật thời gian còn lại

Hình 2 : lưu đồ giải thuật SRTF

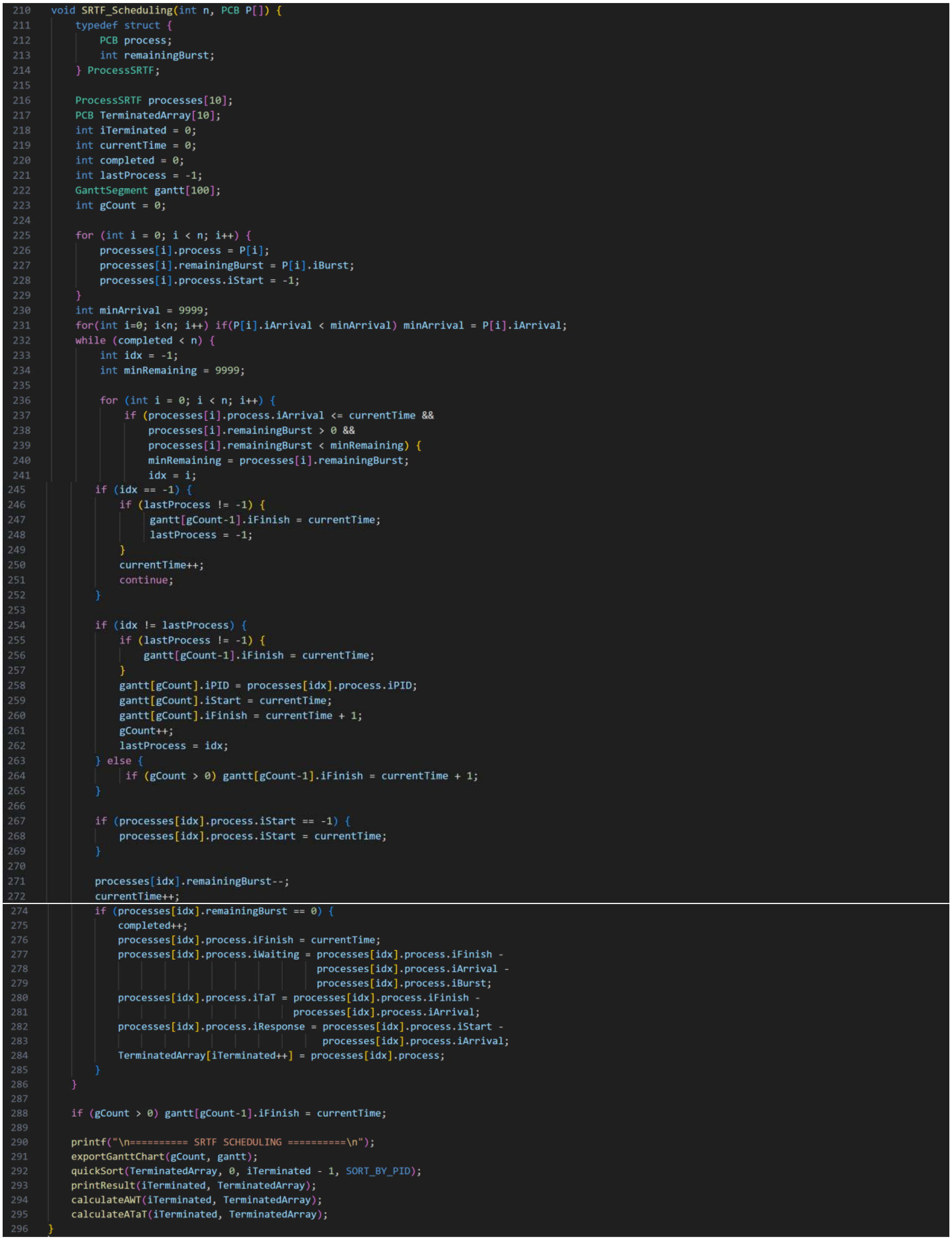
PID	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5
P5	4	2

```
| P1 |   P2   | P5 |   P4   |      P1      |         P3         |
0    1         5    7        12           19                  28
```



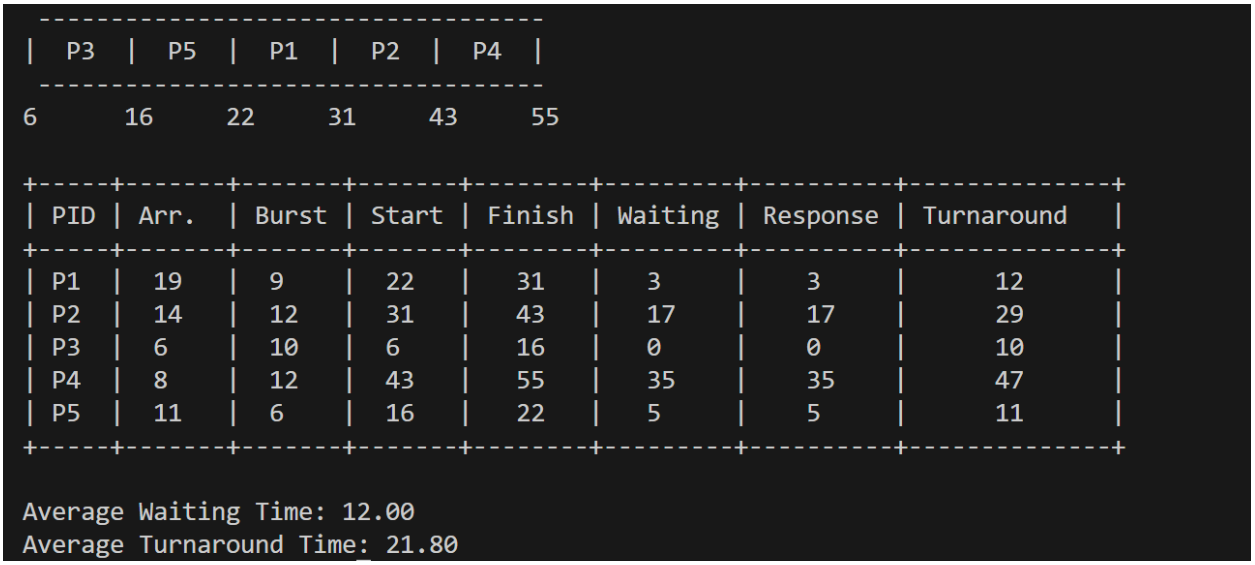
PID	Arrival	Burst	Completion (CT)	Turnaround (TAT	Waiting (WT)
P1	0	8	19	19−0=19	19−8=11
P2	1	4	5	5−1=4	4−4=0
P3	2	9	28	28−2=26	26−9=17
P4	3	5	12	12−3=9	9−5=4
P5	4	2	7	7−4=3	3−2=1

Thời gian chờ trung bình (Average Waiting Time): 6.6
Thời gian lưu lại trung bình (Average Turnaround Time):12.2



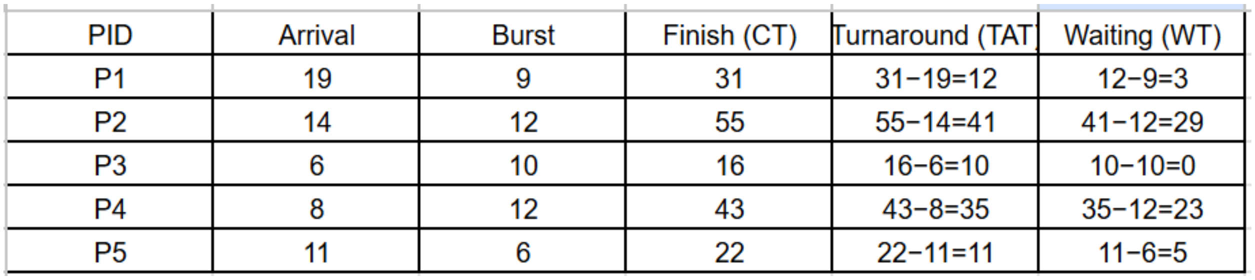
```
void SRTF_Scheduling(int n, PCB P[]) {
    typedef struct {
        PCB process;
        int remainingBurst;
    } ProcessSRTF;

    ProcessSRTF processes[10];
    PCB TerminatedArray[10];
    int iTerminated = 0;
    int currentTime = 0;
    int completed = 0;
    int lastProcess = -1;
    GanttSegment gantt[100];
    int gCount = 0;

    for (int i = 0; i < n; i++) {
        processes[i].process = P[i];
        processes[i].remainingBurst = P[i].iBurst;
        processes[i].process.iStart = -1;
    }
    int minArrival = 9999;
    for(int i=0; i<n; i++) if(P[i].iArrival < minArrival) minArrival = P[i].iArrival;
    while (completed < n) {
        int idx = -1;
        int minRemaining = 9999;

        for (int i = 0; i < n; i++) {
            if (processes[i].process.iArrival <= currentTime &&
                processes[i].remainingBurst > 0 &&
                processes[i].remainingBurst < minRemaining) {
                minRemaining = processes[i].remainingBurst;
                idx = i;
        if (idx == -1) {
            if (lastProcess != -1) {
                gantt[gCount-1].iFinish = currentTime;
                lastProcess = -1;
            }
            currentTime++;
            continue;
        }

        if (idx != lastProcess) {
            if (lastProcess != -1) {
                gantt[gCount-1].iFinish = currentTime;
            }
            gantt[gCount].iPID = processes[idx].process.iPID;
            gantt[gCount].iStart = currentTime;
            gantt[gCount].iFinish = currentTime + 1;
            gCount++;
            lastProcess = idx;
        } else {
            if (gCount > 0) gantt[gCount-1].iFinish = currentTime + 1;
        }

        if (processes[idx].process.iStart == -1) {
            processes[idx].process.iStart = currentTime;
        }

        processes[idx].remainingBurst--;
        currentTime++;
        if (processes[idx].remainingBurst == 0) {
            completed++;
            processes[idx].process.iFinish = currentTime;
            processes[idx].process.iWaiting = processes[idx].process.iFinish -
                                              processes[idx].process.iArrival -
                                              processes[idx].process.iBurst;
            processes[idx].process.iTaT = processes[idx].process.iFinish -
                                          processes[idx].process.iArrival;
            processes[idx].process.iResponse = processes[idx].process.iStart -
                                               processes[idx].process.iArrival;
            TerminatedArray[iTerminated++] = processes[idx].process;
        }
    }

    if (gCount > 0) gantt[gCount-1].iFinish = currentTime;

    printf("\n========== SRTF SCHEDULING ==========\n");
    exportGanttChart(gCount, gantt);
    quickSort(TerminatedArray, 0, iTerminated - 1, SORT_BY_PID);
    printResult(iTerminated, TerminatedArray);
    calculateAWT(iTerminated, TerminatedArray);
    calculateATaT(iTerminated, TerminatedArray);
}
```

--Test 1---



```
 -----------------------------------
| P3  |  P5  |  P1  |  P2  |  P4  |
 -----------------------------------
6      16     22     31     43     55

+-----+-------+-------+-------+--------+---------+----------+--------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround   |
+-----+-------+-------+-------+--------+---------+----------+--------------+
| P1  | 19    | 9     | 22    |  31    | 3       |  3       |      12      |
| P2  | 14    | 12    | 31    |  43    | 17      |  17      |      29      |
| P3  | 6     | 10    | 6     |  16    | 0       |  0       |      10      |
| P4  | 8     | 12    | 43    |  55    | 35      |  35      |      47      |
| P5  | 11    | 6     | 16    |  22    | 5       |  5       |      11      |
+-----+-------+-------+-------+--------+---------+----------+--------------+

Average Waiting Time: 12.00
Average Turnaround Time: 21.80
```

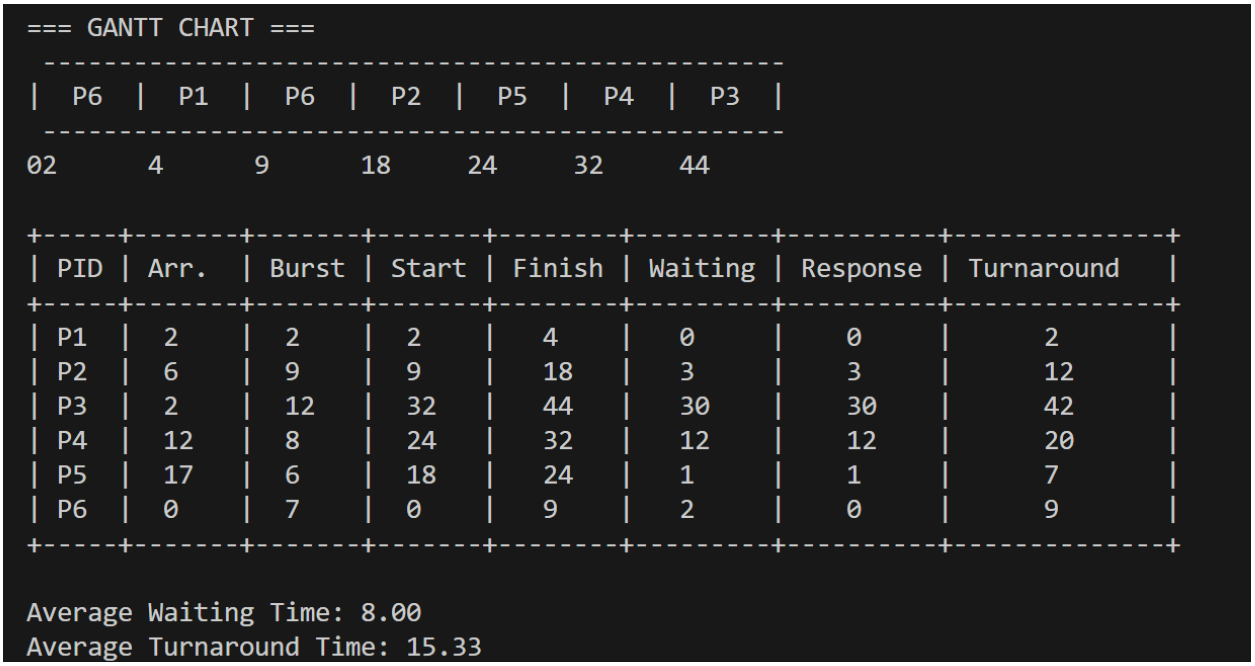
```
|  P3  |  P5  |  P1  |    P2    |    P4    |
6      16     22     31         43         55
```



PID	Arrival	Burst	Finish (CT)	Turnaround (TAT)	Waiting (WT)
P1	19	9	31	31−19=12	12−9=3
P2	14	12	55	55−14=41	41−12=29
P3	6	10	16	16−6=10	10−10=0
P4	8	12	43	43−8=35	35−12=23
P5	11	6	22	22−11=11	11−6=5

Thời gian chờ trung bình (Average Waiting Time): 12.0
Thời gian lưu lại trung bình (Average Turnaround Time): 21.8

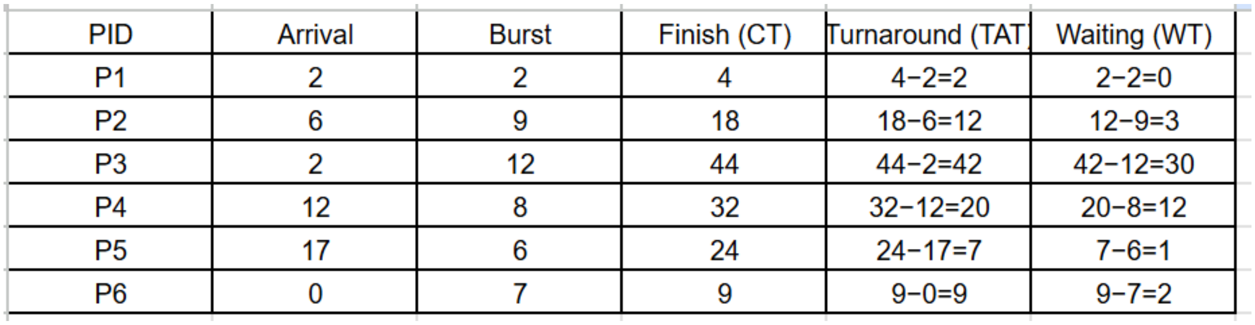
--Test 2---



```
=== GANTT CHART ===
 -------------------------------------------------
|  P6  |  P1  |  P6  |  P2  |  P5  |  P4  |  P3  |
 -------------------------------------------------
02      4      9      18      24      32      44

+-----+-------+-------+-------+--------+---------+----------+---------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround    |
+-----+-------+-------+-------+--------+---------+----------+---------------+
| P1  |  2    |  2    |  2    |   4    |   0     |    0     |      2        |
| P2  |  6    |  9    |  9    |   18   |   3     |    3     |      12       |
| P3  |  2    |  12   |  32   |   44   |   30    |    30    |      42       |
| P4  |  12   |  8    |  24   |   32   |   12    |    12    |      20       |
| P5  |  17   |  6    |  18   |   24   |   1     |    1     |      7        |
| P6  |  0    |  7    |  0    |   9    |   2     |    0     |      9        |
+-----+-------+-------+-------+--------+---------+----------+---------------+

Average Waiting Time: 8.00
Average Turnaround Time: 15.33
```

| P6 | P1 | P6 | P2 | P5 | P4 | P3 |

0   2   4   9   18   24   32   44

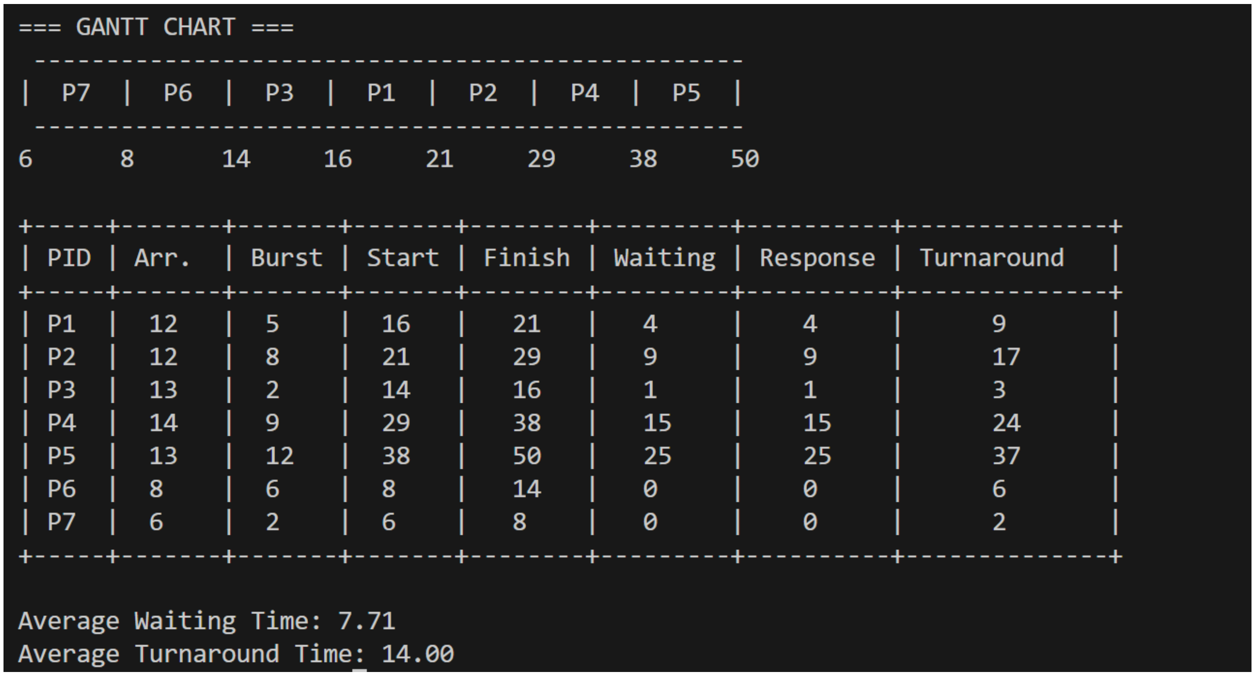


PID	Arrival	Burst	Finish (CT)	Turnaround (TAT	Waiting (WT)
P1	2	2	4	4−2=2	2−2=0
P2	6	9	18	18−6=12	12−9=3
P3	2	12	44	44−2=42	42−12=30
P4	12	8	32	32−12=20	20−8=12
P5	17	6	24	24−17=7	7−6=1
P6	0	7	9	9−0=9	9−7=2

Thời gian chờ trung bình (Average Waiting Time): 8.0

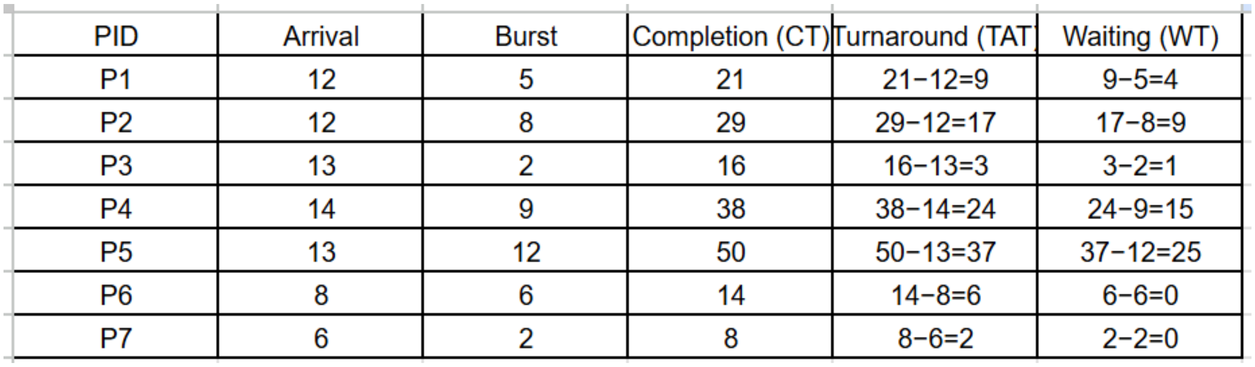
Thời gian lưu lại trung bình (Average Turnaround Time):15.33

---Test 3---



```
=== GANTT CHART ===
 -------------------------------------------------
|  P7  |  P6  |  P3  |  P1  |  P2  |  P4  |  P5  |
 -------------------------------------------------
6      8      14     16     21     29     38     50

+-----+-------+-------+-------+--------+---------+----------+-------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround  |
+-----+-------+-------+-------+--------+---------+----------+-------------+
| P1  |  12   |  5    |  16   |   21   |   4     |    4     |      9      |
| P2  |  12   |  8    |  21   |   29   |   9     |    9     |      17     |
| P3  |  13   |  2    |  14   |   16   |   1     |    1     |      3      |
| P4  |  14   |  9    |  29   |   38   |   15    |    15    |      24     |
| P5  |  13   |  12   |  38   |   50   |   25    |    25    |      37     |
| P6  |  8    |  6    |  8    |   14   |   0     |    0     |      6      |
| P7  |  6    |  2    |  6    |   8    |   0     |    0     |      2      |
+-----+-------+-------+-------+--------+---------+----------+-------------+

Average Waiting Time: 7.71
Average Turnaround Time: 14.00
```

| P7 | P6 | P3 | P1 | P2 | P4 | P5 |
6   8   14   16   21   29   38   50

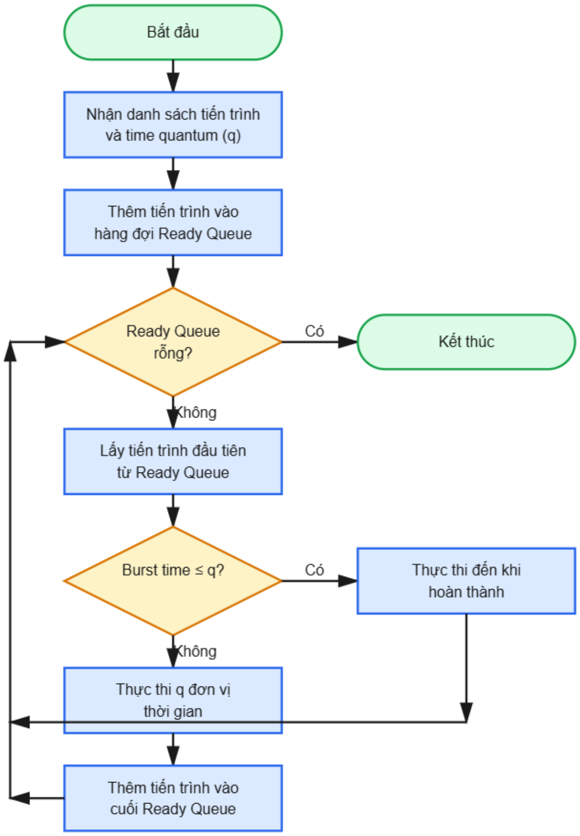


PID	Arrival	Burst	Completion (CT)	Turnaround (TAT)	Waiting (WT)
P1	12	5	21	21−12=9	9−5=4
P2	12	8	29	29−12=17	17−8=9
P3	13	2	16	16−13=3	3−2=1
P4	14	9	38	38−14=24	24−9=15
P5	13	12	50	50−13=37	37−12=25
P6	8	6	14	14−8=6	6−6=0
P7	6	2	8	8−6=2	2−2=0

Thời gian chờ trung bình (Average Waiting Time): 7.71
Thời gian lưu lại trung bình (Average Turnaround Time):14.0

2.6. BÀI TẬP ÔN TẬP

1. Giải thuật Shortest-Remaining-Time-First hoặc Round Robin


Bắt đầu
Nhận danh sách tiến trình và time quantum (q)
Thêm tiến trình vào hàng đợi Ready Queue
Ready Queue rỗng?
Có
Kết thúc
Không
Lấy tiến trình đầu tiên từ Ready Queue
Burst time ≤ q?
Có
Thực thi đến khi hoàn thành
Không
Thực thi q đơn vị thời gian
Thêm tiến trình vào cuối Ready Queue

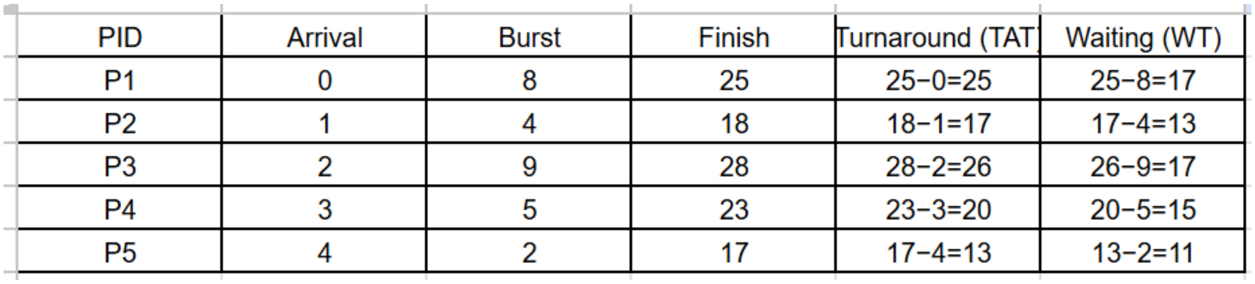
Hình 3: Lưu đồ giải thuật RR

PID	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5
P5	4	2

Giả sử q = 3:

| P1 | P2 | P3 | P4 | P1 | P5 | P2| P3 | P4| P1| P3 |

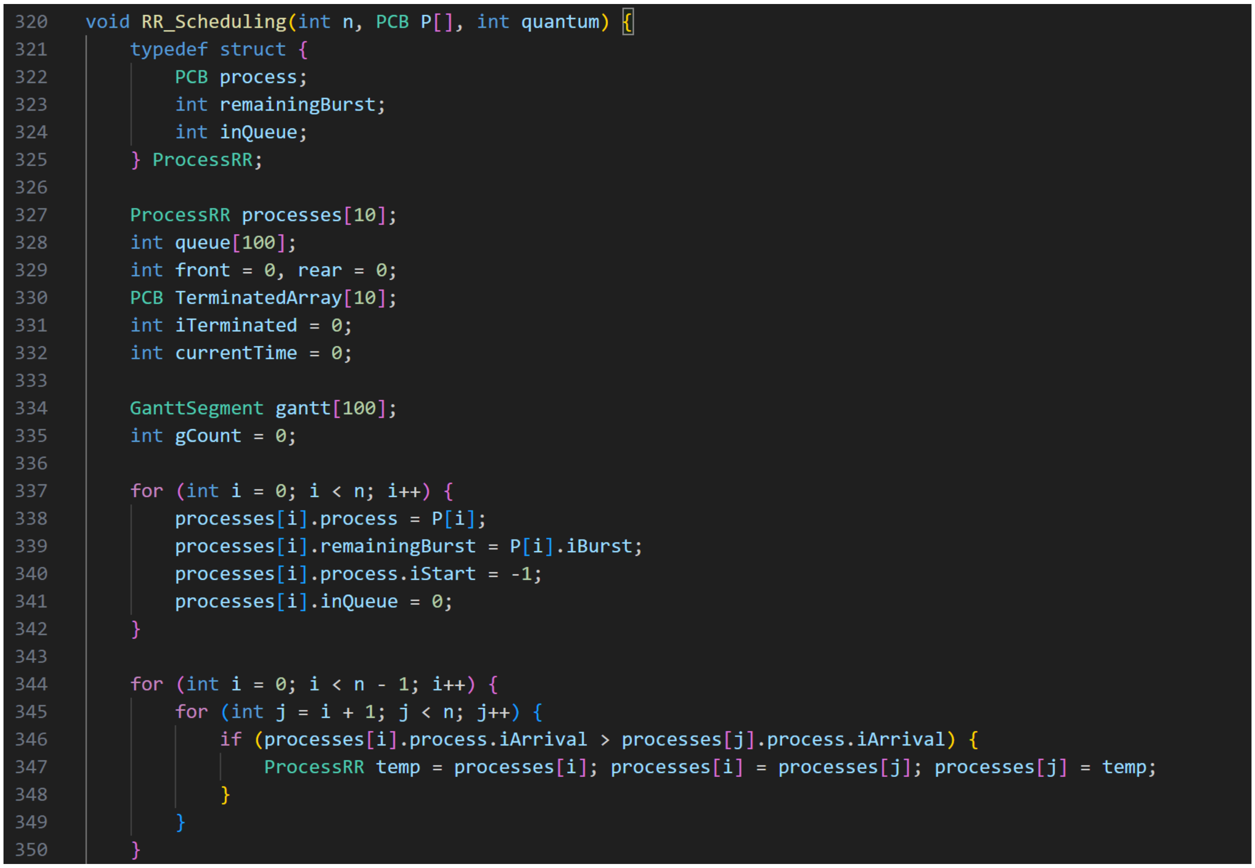
0 3 6 9 12 15 17 18 21 23 25 28



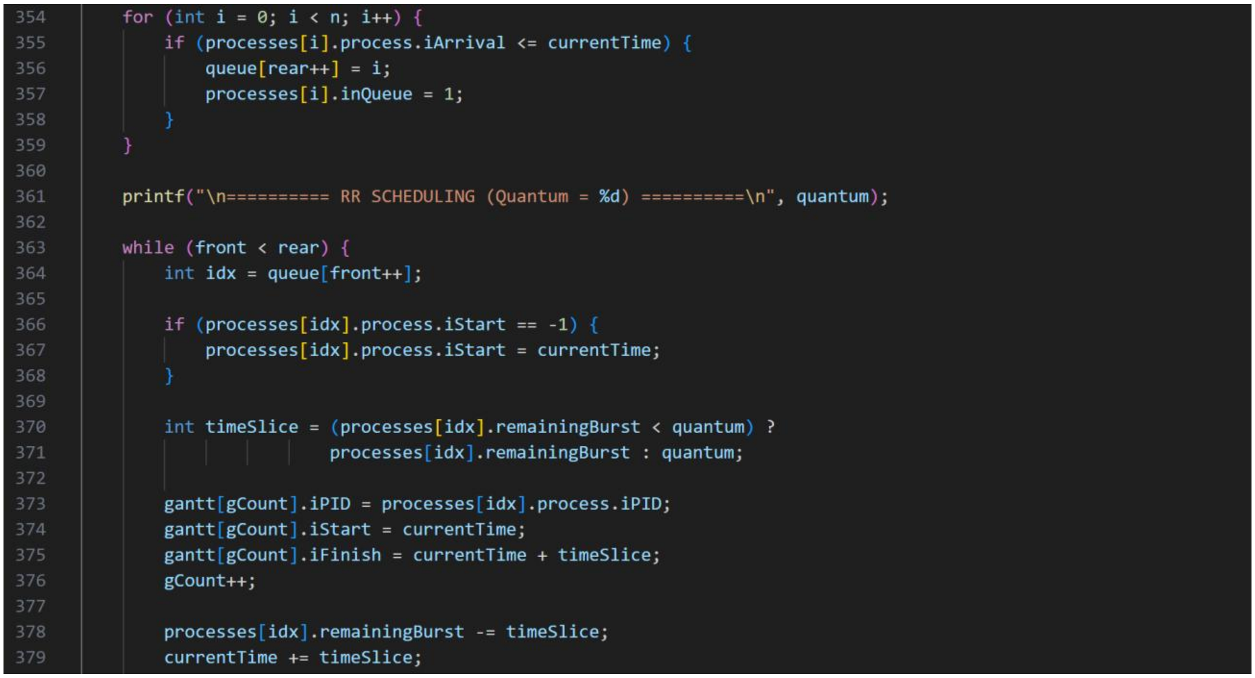
PID	Arrival	Burst	Finish	Turnaround (TAT	Waiting (WT)
P1	0	8	25	25−0=25	25−8=17
P2	1	4	18	18−1=17	17−4=13
P3	2	9	28	28−2=26	26−9=17
P4	3	5	23	23−3=20	20−5=15
P5	4	2	17	17−4=13	13−2=11

Thời gian chờ trung bình (Average Waiting Time): 14.6

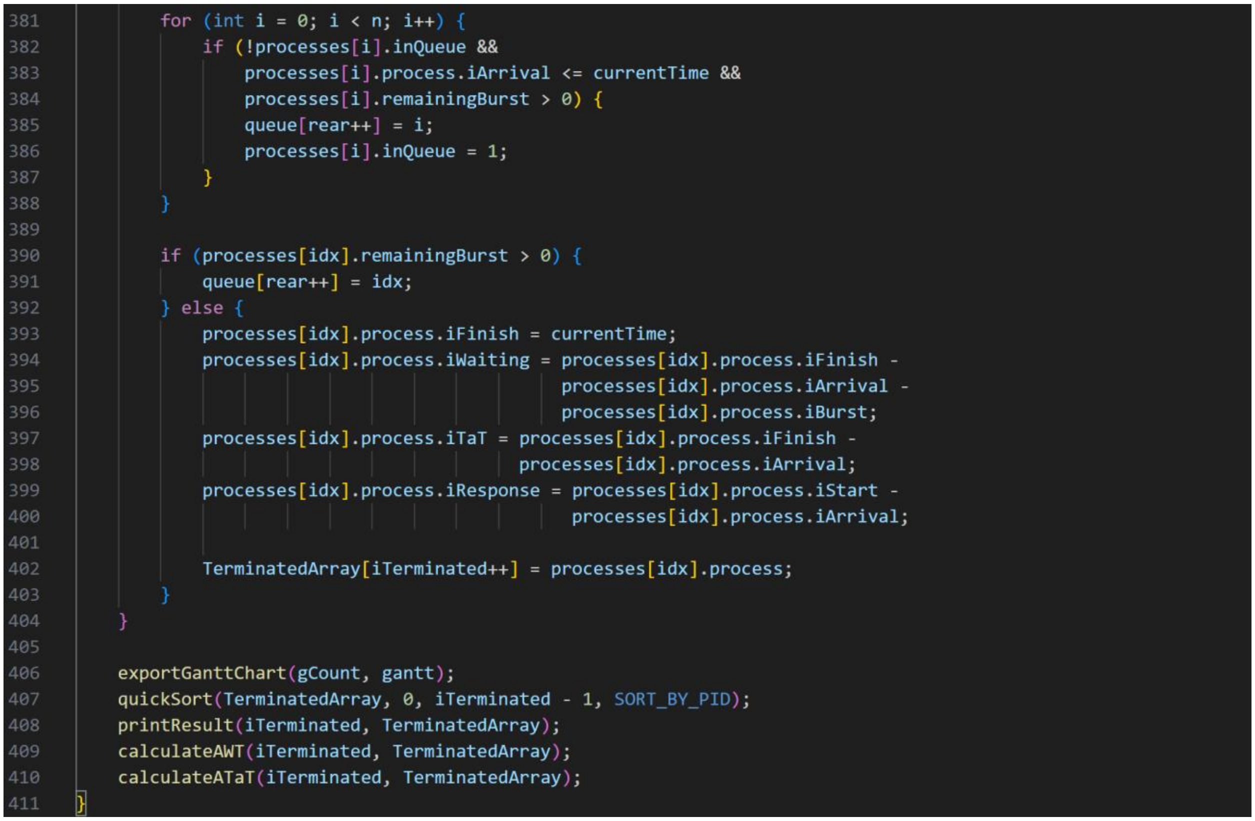
Thời gian lưu lại trung bình (Average Turnaround Time): 20.2



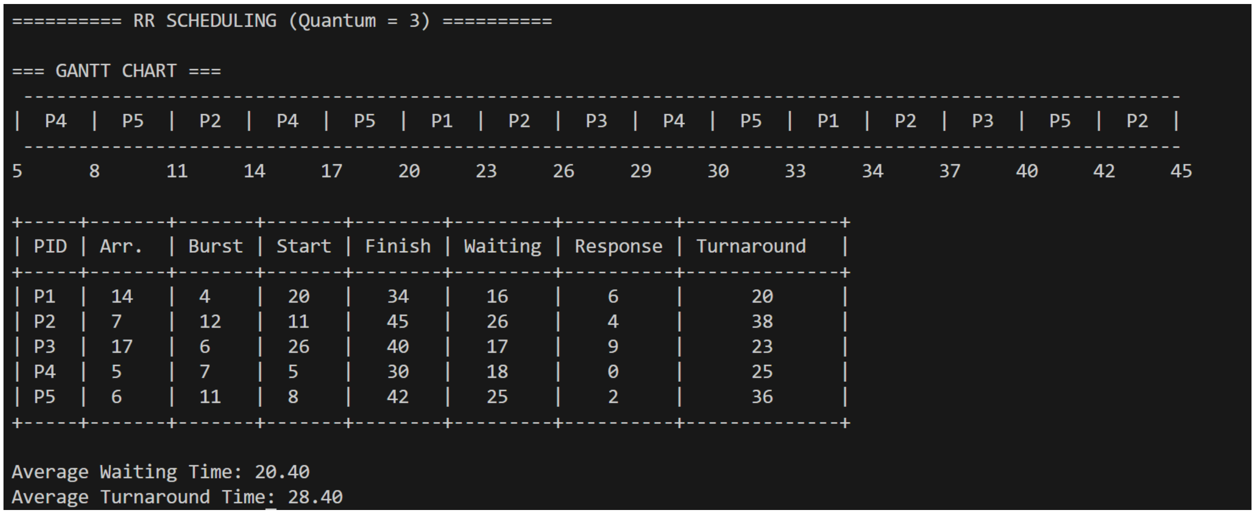
```
void RR_Scheduling(int n, PCB P[], int quantum) {
    typedef struct {
        PCB process;
        int remainingBurst;
        int inQueue;
    } ProcessRR;

    ProcessRR processes[10];
    int queue[100];
    int front = 0, rear = 0;
    PCB TerminatedArray[10];
    int iTerminated = 0;
    int currentTime = 0;

    GanttSegment gantt[100];
    int gCount = 0;

    for (int i = 0; i < n; i++) {
        processes[i].process = P[i];
        processes[i].remainingBurst = P[i].iBurst;
        processes[i].process.iStart = -1;
        processes[i].inQueue = 0;
    }

    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (processes[i].process.iArrival > processes[j].process.iArrival) {
                ProcessRR temp = processes[i]; processes[i] = processes[j]; processes[j] = temp;
            }
        }
    }
```



```c
    for (int i = 0; i < n; i++) {
        if (processes[i].process.iArrival <= currentTime) {
            queue[rear++] = i;
            processes[i].inQueue = 1;
        }
    }

    printf("\n========== RR SCHEDULING (Quantum = %d) ==========\n", quantum);

    while (front < rear) {
        int idx = queue[front++];

        if (processes[idx].process.iStart == -1) {
            processes[idx].process.iStart = currentTime;
        }

        int timeSlice = (processes[idx].remainingBurst < quantum) ?
                        processes[idx].remainingBurst : quantum;

        gantt[gCount].iPID = processes[idx].process.iPID;
        gantt[gCount].iStart = currentTime;
        gantt[gCount].iFinish = currentTime + timeSlice;
        gCount++;

        processes[idx].remainingBurst -= timeSlice;
        currentTime += timeSlice;
```



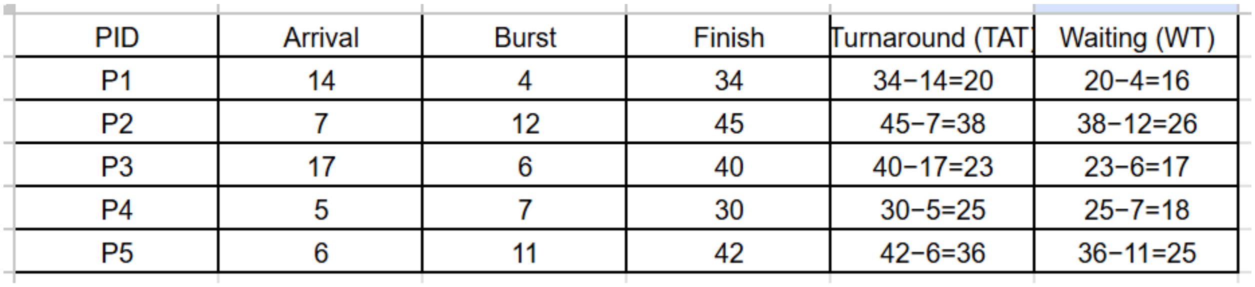
```c
        for (int i = 0; i < n; i++) {
            if (!processes[i].inQueue &&
                processes[i].process.iArrival <= currentTime &&
                processes[i].remainingBurst > 0) {
                queue[rear++] = i;
                processes[i].inQueue = 1;
            }
        }

        if (processes[idx].remainingBurst > 0) {
            queue[rear++] = idx;
        } else {
            processes[idx].process.iFinish = currentTime;
            processes[idx].process.iWaiting = processes[idx].process.iFinish -
                                              processes[idx].process.iArrival -
                                              processes[idx].process.iBurst;
            processes[idx].process.iTaT = processes[idx].process.iFinish -
                                          processes[idx].process.iArrival;
            processes[idx].process.iResponse = processes[idx].process.iStart -
                                               processes[idx].process.iArrival;

            TerminatedArray[iTerminated++] = processes[idx].process;
        }
    }

    exportGanttChart(gCount, gantt);
    quickSort(TerminatedArray, 0, iTerminated - 1, SORT_BY_PID);
    printResult(iTerminated, TerminatedArray);
    calculateAWT(iTerminated, TerminatedArray);
    calculateATaT(iTerminated, TerminatedArray);
}
```



```
========== RR SCHEDULING (Quantum = 3) ==========

=== GANTT CHART ===
 ----------------------------------------------------------------------------------------------------------------
|  P4  |  P5  |  P2  |  P4  |  P5  |  P1  |  P2  |  P3  |  P4  |  P5  |  P1  |  P2  |  P3  |  P5  |  P2  |
 ----------------------------------------------------------------------------------------------------------------
5      8      11     14     17     20     23     26     29     30     33     34     37     40     42     45

+-----+-------+-------+-------+--------+---------+----------+--------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround   |
+-----+-------+-------+-------+--------+---------+----------+--------------+
| P1  | 14    | 4     | 20    | 34     | 16      | 6        |     20       |
| P2  | 7     | 12    | 11    | 45     | 26      | 4        |     38       |
| P3  | 17    | 6     | 26    | 40     | 17      | 9        |     23       |
| P4  | 5     | 7     | 5     | 30     | 18      | 0        |     25       |
| P5  | 6     | 11    | 8     | 42     | 25      | 2        |     36       |
+-----+-------+-------+-------+--------+---------+----------+--------------+

Average Waiting Time: 20.40
Average Turnaround Time: 28.40
```

| P4 | P5 | P2 | P4 | P5 | P1 | P2 | P3 | P4| P5 | P1| P2 | P3 | P5| P2 |

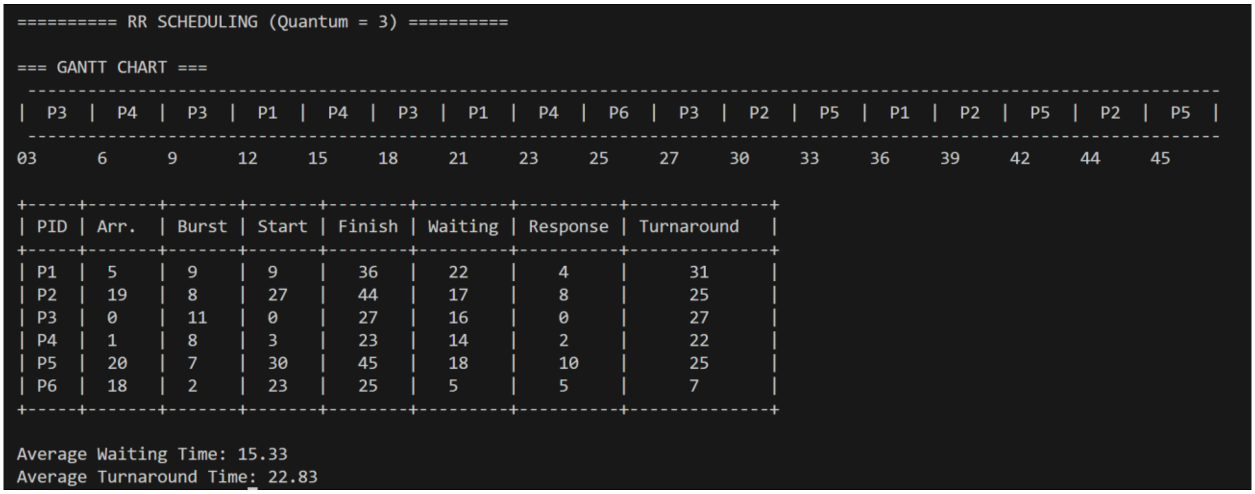
5   8   11   14   17   20   23   26   29 30   33 34   37   40 42   45



PID	Arrival	Burst	Finish	Turnaround (TAT	Waiting (WT)
P1	14	4	34	34−14=20	20−4=16
P2	7	12	45	45−7=38	38−12=26
P3	17	6	40	40−17=23	23−6=17
P4	5	7	30	30−5=25	25−7=18
P5	6	11	42	42−6=36	36−11=25

Thời gian chờ trung bình (Average Waiting Time): 20.4

Thời gian lưu lại trung bình (Average Turnaround Time): 28.4



```
========== RR SCHEDULING (Quantum = 3) ==========

=== GANTT CHART ===
 ----------------------------------------------------------------------------------------------------------------------------
|  P3  |  P4  |  P3  |  P1  |  P4  |  P3  |  P1  |  P4  |  P6  |  P3  |  P2  |  P5  |  P1  |  P2  |  P5  |  P2  |  P5  |
 ----------------------------------------------------------------------------------------------------------------------------
03     6      9      12     15     18     21     23     25     27     30     33     36     39     42     44     45

+-----+-------+-------+-------+--------+---------+----------+--------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround   |
+-----+-------+-------+-------+--------+---------+----------+--------------+
| P1  | 5     | 9     | 9     |  36    | 22      |    4     |     31       |
| P2  | 19    | 8     | 27    |  44    | 17      |    8     |     25       |
| P3  | 0     | 11    | 0     |  27    | 16      |    0     |     27       |
| P4  | 1     | 8     | 3     |  23    | 14      |    2     |     22       |
| P5  | 20    | 7     | 30    |  45    | 18      |    10    |     25       |
| P6  | 18    | 2     | 23    |  25    | 5       |    5     |     7        |
+-----+-------+-------+-------+--------+---------+----------+--------------+

Average Waiting Time: 15.33
Average Turnaround Time: 22.83
```

| P3 | P4 | P3 | P1 | P4 | P3 | P1 | P4 | P6 | P3 | P2 | P5 | P1 | P2 | P5 | P2 | P5|

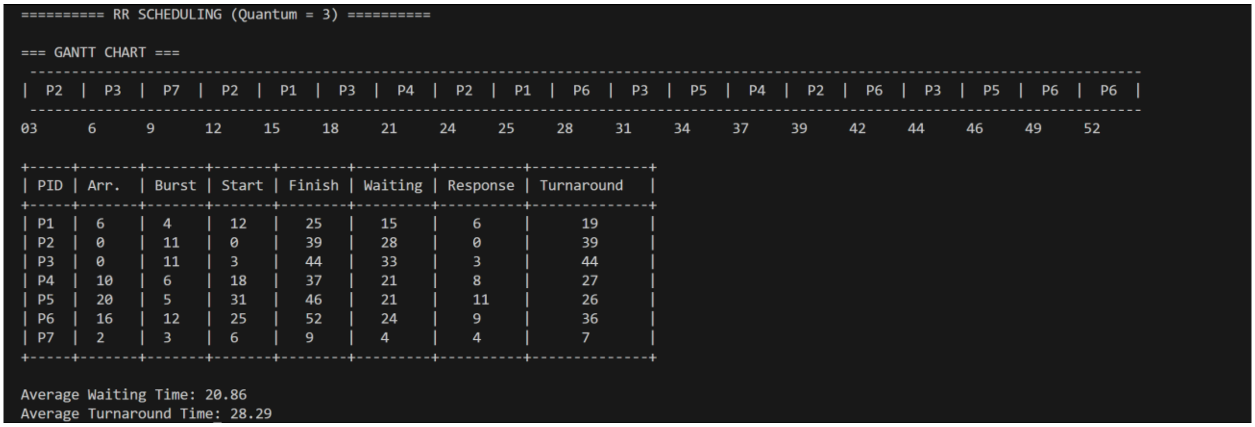
0   3   6   9   12   15   18   21   23   25   27   30   33   36   39   42   44 45

PID	Arrival	Burst	Finish	Turnaround (TAT)	Waiting (WT)
P1	5	9	36	36−5=31	31−9=22
P2	19	8	44	44−19=25	25−8=17
P3	0	11	27	27−0=27	27−11=16
P4	1	8	23	23−1=22	22−8=14
P5	20	7	45	45−20=25	25−7=18
P6	18	2	25	25−18=7	7−2=5

Thời gian chờ trung bình (Average Waiting Time): 15.33

Thời gian lưu lại trung bình (Average Turnaround Time): 22.83

Test 3:



```
========== RR SCHEDULING (Quantum = 3) ==========

=== GANTT CHART ===
 ---------------------------------------------------------------------------------------------------------------------------------------------
| P2 | P3 | P7 | P2 | P1 | P3 | P4 | P2 | P1 | P6 | P3 | P5 | P4 | P2 | P6 | P3 | P5 | P6 | P6 |
 ---------------------------------------------------------------------------------------------------------------------------------------------
03     6      9      12     15     18     21     24     25     28     31     34     37     39     42     44     46     49     52

+-----+-------+-------+-------+--------+---------+----------+-------------+
| PID | Arr.  | Burst | Start | Finish | Waiting | Response | Turnaround  |
+-----+-------+-------+-------+--------+---------+----------+-------------+
| P1  | 6     | 4     | 12    |   25   |   15    |    6     |     19      |
| P2  | 0     | 11    | 0     |   39   |   28    |    0     |     39      |
| P3  | 0     | 11    | 3     |   44   |   33    |    3     |     44      |
| P4  | 10    | 6     | 18    |   37   |   21    |    8     |     27      |
| P5  | 20    | 5     | 31    |   46   |   21    |    11    |     26      |
| P6  | 16    | 12    | 25    |   52   |   24    |    9     |     36      |
| P7  | 2     | 3     | 6     |   9    |   4     |    4     |     7       |
+-----+-------+-------+-------+--------+---------+----------+-------------+

Average Waiting Time: 20.86
Average Turnaround Time: 28.29
```

| P2 | P3 | P7 | P2 | P1 | P3 | P4 | P2 | P1| P6 | P3 | P5 | P4 | P2| P6 | P3| P5| P6 | P6 |

0   3   6   9   12   15   18   21   24 25   28   31   34   37 39   42 44 46   49   52

PID	Arrival	Burst	Finish	Turnaround (TAT)	Waiting (WT)
P1	6	4	25	25−6=19	19−4=15
P2	0	11	39	39−0=39	39−11=28
P3	0	11	44	44−0=44	44−11=33
P4	10	6	37	37−10=27	27−6=21
P5	20	5	46	46−20=26	26−5=21
P6	16	12	52	52−16=36	36−12=24
P7	2	3	9	9−2=7	7−3=4

Thời gian chờ trung bình (Average Waiting Time): 20.86

Thời gian lưu lại trung bình (Average Turnaround Time): 28.29