

Name: Nguyễn Huỳnh Nhân

ID:23521080

Class:IT007.P28.2

## OPERATING SYSTEM LAB X'S REPORT

### SUMMARY

Task		Status	Page
Section 4.5	Ex 1	Hoàn thành	1
	Ex 2	Hoàn thành	7
...	...		
	...		

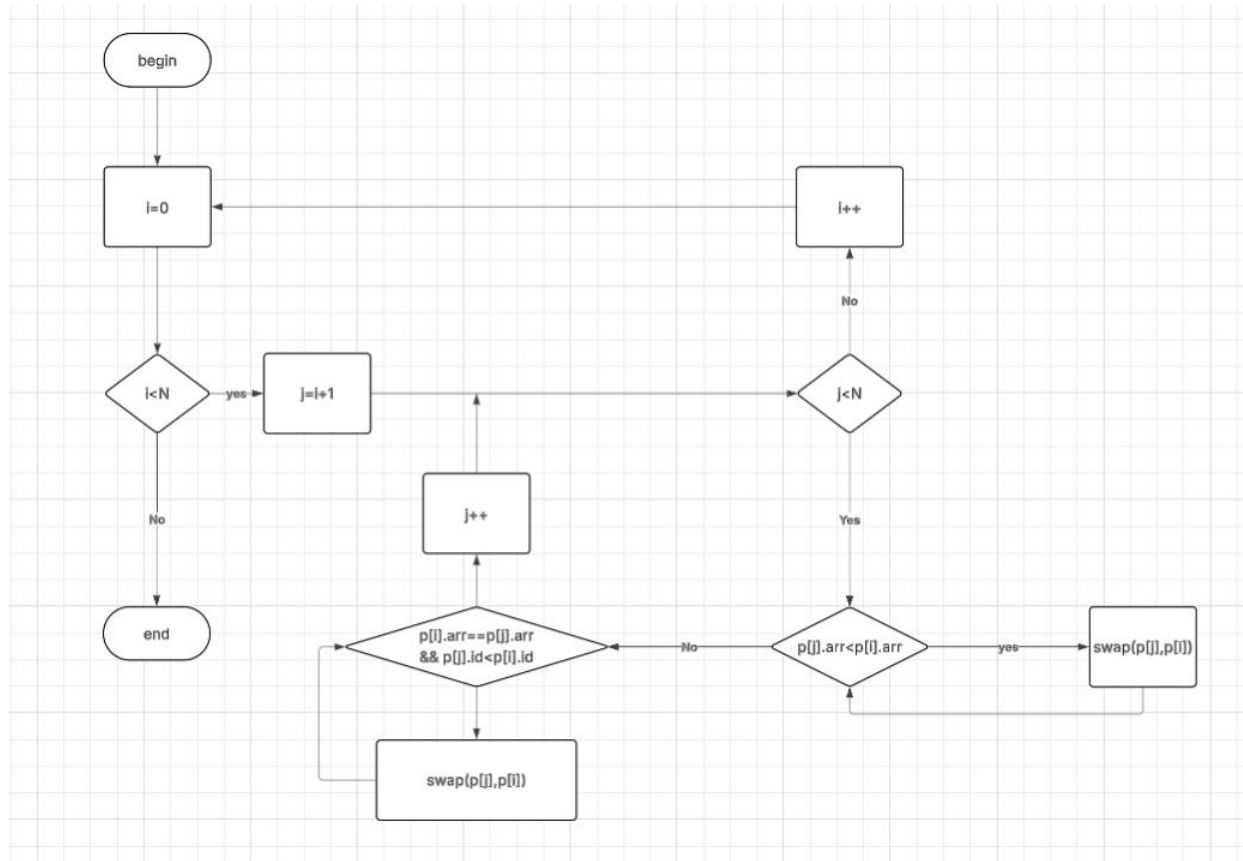
Self-scores: 10/10

*\*Note: Export file to **PDF** and name the file by following format:  
**Student ID\_LABx.pdf***

## Section 4.5

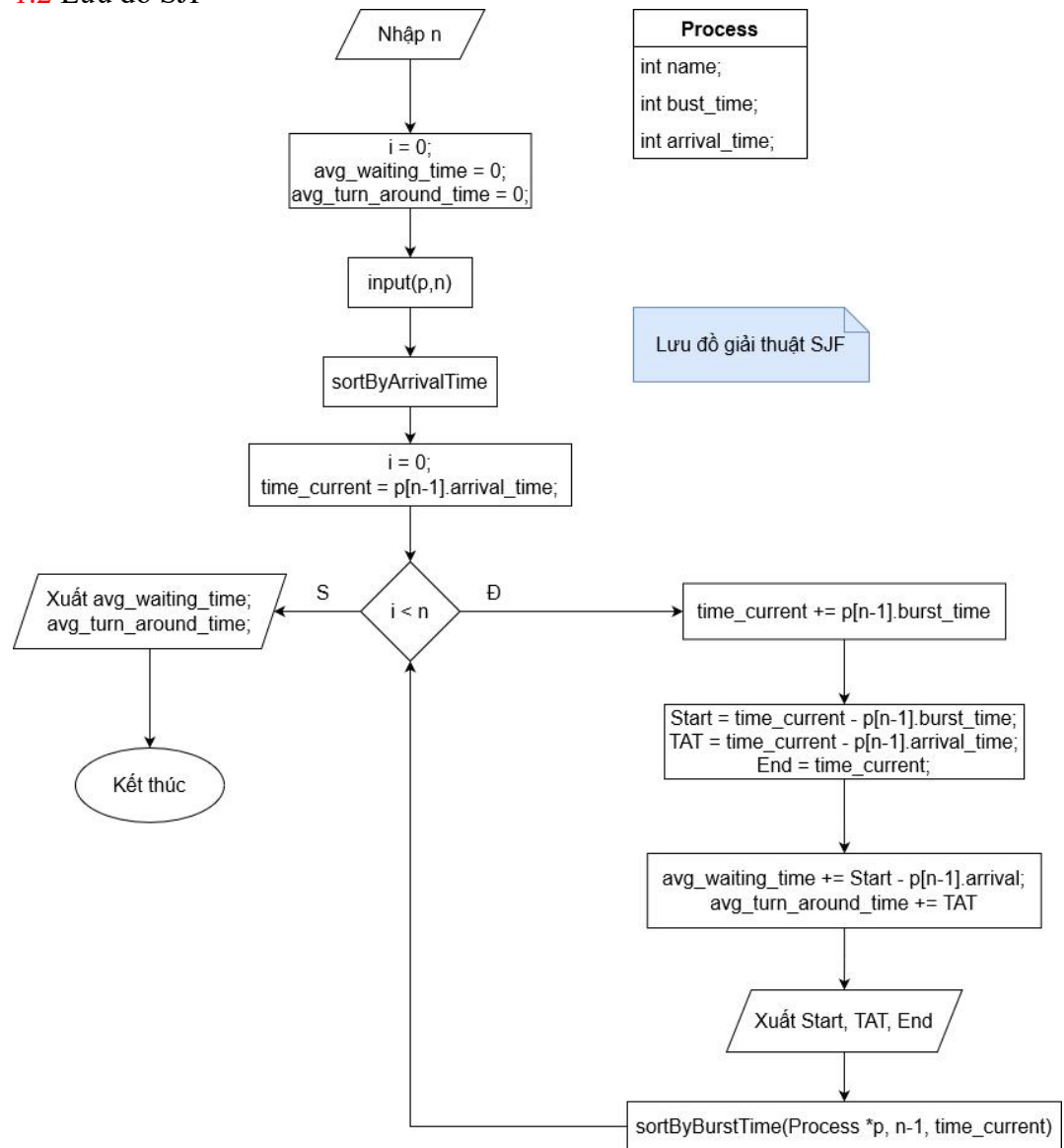
### 1. Task name 1: Viết chương trình mô phỏng SJF

#### 1.1 Hàm sort theo arrival time



Giải thích :sử dụng thuật toán nổi bọt để sắp xếp tăng dần theo arrival time

## 1.2 Lưu đồ SJF



## 1.3 Code

```

void SJF(struct process P[]){

    int done[num_process];
    for (int i = 0; i < num_process; i++) done[i] = 0;

    float time_elapsed = P[0].arr + P[0].brust;
    P[0].finish = time_elapsed;
    P[0].turnround_time = P[0].finish - P[0].arr;
    P[0].wait_time = P[0].turnround_time - P[0].brust;
    P[0].rp_time = P[0].wait_time;

    done[0] = 1;
    for (int i = 1; i < num_process; i++){
        //printf("%d %.2f\n",i,time_elapsed);
        float shortestBrust = INF;
        int idx = -1;
        for (int j = 0; j < num_process; j++){
            if (!done[j] && time_elapsed >= P[j].arr && shortestBrust > P[j].brust){
                shortestBrust = P[j].brust;
                idx = j;
            }
        }
        if (idx == -1){
            float shortestarr = INF;
            for (int j = 0; j < num_process; j++){
                if (!done[j] && shortestarr > P[j].arr && P[j].arr > time_elapsed) {
                    shortestarr = P[j].arr;
                    idx = j;
                }
            }
            if (idx == -1) break;
            time_elapsed = shortestarr;
        }
        time_elapsed += P[idx].brust;
        P[idx].finish = time_elapsed;
        P[idx].turnround_time = P[idx].finish - P[idx].arr;
        P[idx].wait_time = P[idx].turnround_time - P[idx].brust;
        P[idx].rp_time = P[idx].wait_time;
        done[idx] = 1;
    }
}

```

#### 1.4 Test case

Process l	Arrival Time	Burst Time
P1	0	9
P2	4	5
P3	2	7
P4	8	10
P5	10	13

## Chạy code

Process Name	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time
p1	0.000000	9.000000	9.000000	9.000000	0.000000	0.000000
p2	4.000000	5.000000	14.000000	10.000000	5.000000	5.000000
p3	2.000000	7.000000	21.000000	19.000000	12.000000	12.000000
p4	8.000000	10.000000	31.000000	23.000000	13.000000	13.000000
p5	10.000000	13.000000	44.000000	34.000000	21.000000	21.000000

Average waiting time : 10.20  
 Average turn around time: 19.00  
 Average response time: 10.20  
 Enter algorithm:

## Chạy tay

+ Giải đồ Gantt:

P1	P2	P3	P4	P5
0	9	14	21	31
				44

+ Thời gian đáp ứng:

P1 = 0, P2 = 5, P3 = 12, P4 = 13, P5 = 21

⇒ Thời gian đáp ứng trung bình:  $(0 + 5 + 12 + 13 + 21) / 5 = 10.2$

+ Thời gian đợi:

P1 = 0, P2 = 5, P3 = 12, P4 = 13, P5 = 21

⇒ Thời gian đợi trung bình:  $(0 + 5 + 12 + 13 + 21) / 5 = 10.2$

+ Thời gian hoàn thành:

P1 = 9, P2 = 10, P3 = 19, P4 = 23, P5 = 34

⇒ Thời gian hoàn thành trung bình:  $(9 + 10 + 19 + 23 + 34) / 5 = 19$

## Task2

```

Enter number of process: 5
Enter Process Name: p1
Enter Arrival Time: 0
Enter Burst Time: 12
Enter Process Name: 2
Enter Arrival Time: 2
Enter Burst Time: 7
Enter Process Name: 3
Enter Arrival Time: 5
Enter Burst Time: 8
Enter Process Name: 4
Enter Arrival Time: 9
Enter Burst Time: 3
Enter Process Name: 5
Enter Arrival Time: 12
Enter Burst Time: 6

Enter algorithm (SJF, SRTF, RR, FCFS or EXIT): SJF

```

Process Name	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time
p1	0.00	12.00	12.00	12.00	0.00	0.00
2	2.00	7.00	28.00	26.00	19.00	19.00
3	5.00	8.00	36.00	31.00	23.00	23.00
4	9.00	3.00	15.00	6.00	3.00	3.00
5	12.00	6.00	21.00	9.00	3.00	3.00

```

Average waiting time : 9.60
Average turn around time: 16.80
Average response time: 9.60

```

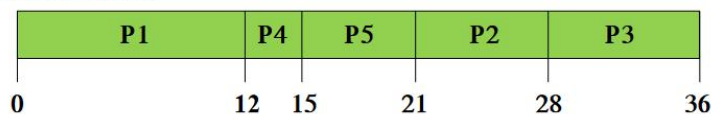
```

Enter algorithm (SJF, SRTF, RR, FCFS or EXIT): █

```

Chạy tay

#### ■ Giản đồ Gantt



#### ■ Thời gian chờ:

- ▣  $P1 = 0, P2 = 19, P3 = 23, P4 = 3, P5 = 3$
- ▣ Thời gian chờ trung bình:  $(0 + 19 + 23 + 3 + 3)/5 = 9.6$

#### ■ Thời gian đáp ứng:

- ▣  $P1 = 0, P2 = 19, P3 = 23, P4 = 3, P5 = 3$
- ▣ Thời gian đáp ứng trung bình:  $(0 + 19 + 23 + 3 + 3)/5 = 9.6$

#### ■ Thời gian hoàn thành:

- ▣  $P1 = 12, P2 = 26, P3 = 31, P4 = 6, P5 = 9$
- ▣ Thời gian hoàn thành trung bình:  $(12 + 26 + 31 + 6 + 9)/5 = 16.8$

### TASK3

Enter algorithm (SJF, SRTF, RR, FCFS or EXIT): SJF

Process Name	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time
1	0.00	8.00	8.00	8.00	0.00	0.00
2	2.00	19.00	48.00	46.00	27.00	27.00
3	4.00	3.00	11.00	7.00	4.00	4.00
4	5.00	6.00	17.00	12.00	6.00	6.00
5	7.00	12.00	29.00	22.00	10.00	10.00

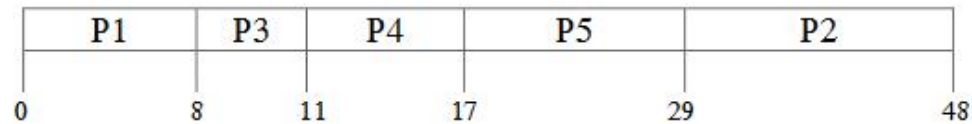
Average waiting time : 9.40

Average turn around time: 19.00

Average response time: 9.40

Chạy tay

+ Giản đồ Gantt:



+ Thời gian đáp ứng trung bình là: 9.4

+ Thời gian hoàn thành trung bình: 19.

## 2. Task name 2: giải thuật RR

### a. Giải thuật RR

#### – Giải thích:

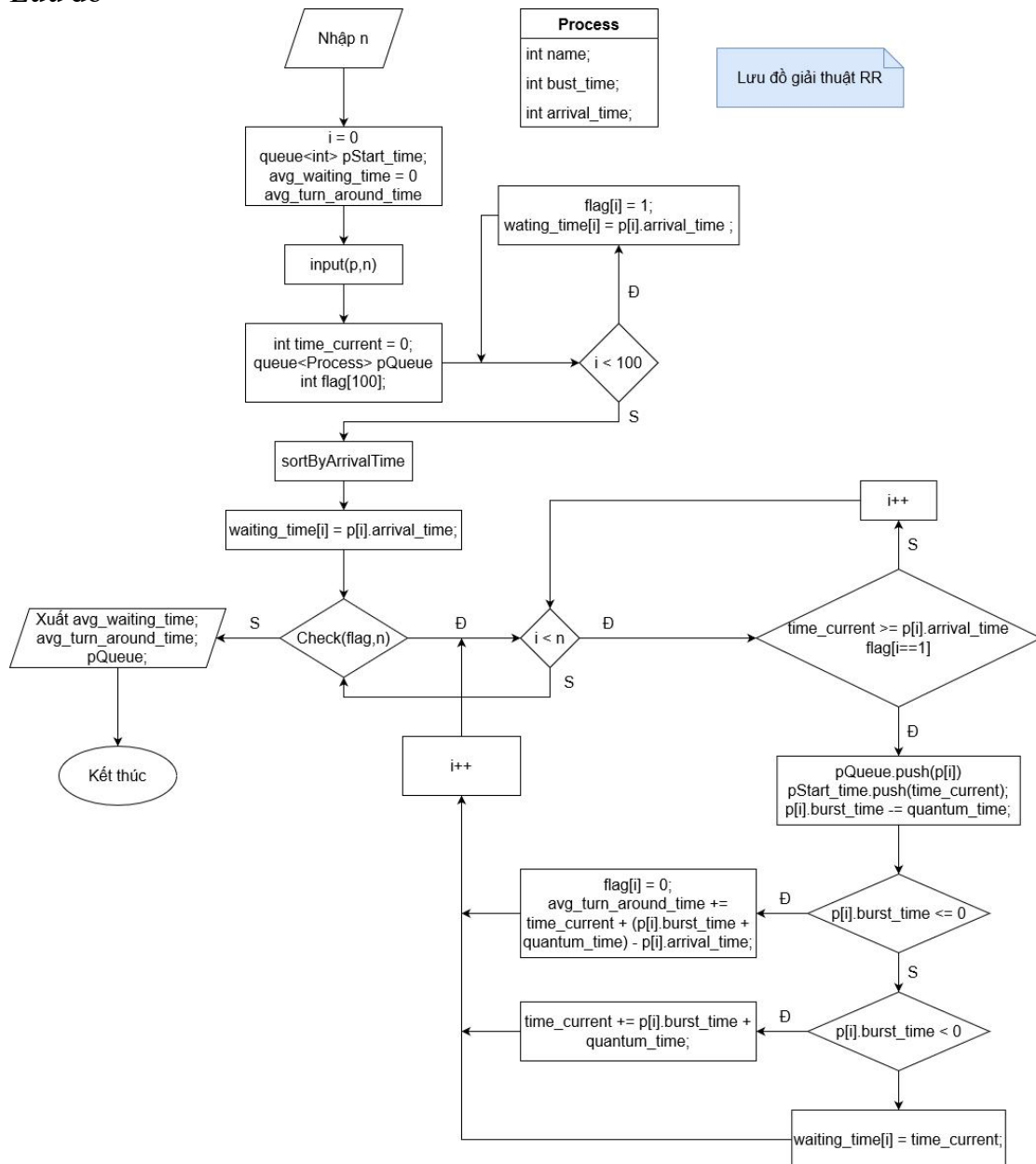
- Tương tự như các giải thuật trên ta sẽ tiến hành tạo struct và tiến hành nhập các thông tin. Và nhập quantumtime.
- Tạo list flag có tác dụng kiểm tra xem burst\_time của tiến trình còn hay không. Ban đầu ta sẽ gán hết bằng 1. Và waiting\_time sẽ bằng thời gian đến.
- Tiến hành sort theo arrival time. Và tạo ra 1 list các tiến trình rỗng khác
- Dùng hàm check để kiểm tra xem còn tiến trình nào vẫn còn burst\_time hay không.
- Duyệt qua lần lượt tất cả các process. Tiến trình nào đã đến và còn burst time thì được xét. Ta sẽ tạo bản sao và được gán vào list qQueue và các thông tin khác sẽ được lưu. Sau đó giảm burstTime đi với số lượng = quantum\_time
- Còn nếu burst\_time == 0 thì cho flag[i] = 0 và tính thời gian đợi và thời gian hoàn thành. Nếu bursttime < 0 thì thời timeline lúc này



sẽ được cộng thêm burst\_time và quantum\_time. Các trường hợp khác thì time\_current là timeline lúc này được cộng thêm quantum\_time.

- Tiến hành với các tiến trình khác và kiểm tra day flag còn phần tử nào bằng 1 hay không. Nếu hết rồi thì xuất ra qQueue. Từ đó sẽ lấy được thông tin cụ thể.

Lưu đồ



**B Code**

```
void RR(struct process P[]) {
```



```
float quantum;  
printf("Enter Quantum time: ");  
scanf("%f", &quantum);
```

```
pQueue q = createQueue();  
float time_elapsed = 0;  
int done[N] = {0};
```

```
int numP = 0;  
int cntP = num_process;
```

```
while (cntP > 0) {  
    while (numP < num_process && P[numP].arr <=  
time_elapsed) {  
        push(q, numP++);  
    }  
    if (isEmpty(q)) {  
        if (numP < num_process)  
            time_elapsed = P[numP].arr;  
        continue;  
    }  
    int idx = front(q);  
    pop(q);
```

```
    if (P[idx].remain_time == P[idx].brust)  
        P[idx].rp_time = time_elapsed - P[idx].arr;
```

```
    float rtime = (P[idx].remain_time < quantum) ?  
P[idx].remain_time : quantum;  
    time_elapsed += rtime;  
    P[idx].remain_time -= rtime;
```

```
    while (numP < num_process && P[numP].arr <=  
time_elapsed) {  
        push(q, numP++);  
    }
```

```
    if (P[idx].remain_time == 0) {
```

```

        P[idx].finish = time_elapsed;
        P[idx].turnround_time = P[idx].finish - P[idx].arr;
        P[idx].wait_time = P[idx].turnround_time -
P[idx].brust;
        done[idx] = 1;
        cntP--;
    } else {
        push(q, idx);
    }
}
freeQueue(q);
show(P);
}

```

## C Test case

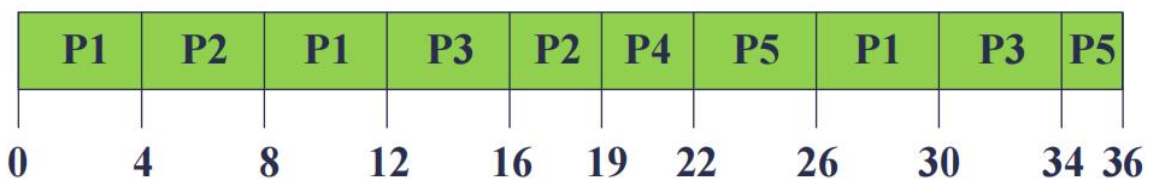
Enter algorithm (SJF, SRTF, RR, FCFS or EXIT): RR  
Enter Quantum time: 4

Process Name	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time
1	0.00	12.00	30.00	30.00	18.00	0.00
2	2.00	7.00	19.00	17.00	10.00	2.00
3	5.00	8.00	34.00	29.00	21.00	7.00
4	9.00	3.00	22.00	13.00	10.00	10.00
5	12.00	6.00	36.00	24.00	18.00	10.00

Average waiting time : 15.40  
Average turn around time: 22.60  
Average response time: 5.80

Enter algorithm (SJF, SRTF, RR, FCFS or EXIT):

Giải tay:



Thời gian đáp ứng trung bình: 5.8

Thời gian chờ trung bình: 15.4

Thời gian hoàn thành trung bình: 22.6

Ví dụ 2:

```
Enter algorithm (SJF, SRTF, RR, FCFS or EXIT): RR
Enter Quantum time: 6
```

Process Name	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time
1	2.00	11.00	30.00	28.00	17.00	4.00
2	0.00	8.00	14.00	14.00	6.00	0.00
3	7.00	5.00	19.00	12.00	7.00	7.00
4	9.00	10.00	40.00	31.00	21.00	10.00
5	13.00	6.00	36.00	23.00	17.00	17.00

Average waiting time : 13.60  
Average turn around time: 21.60  
Average response time: 7.60

Chạy tay:

...

P2	P1	P2	P3	P4	P1	P5	P4	
0	6	12	14	19	25	30	36	40

Thời gian đợi trung bình:  $((4 + 13) + 6 + 7 + (10 + 11) + 17)/5 = 13.6$

Thời gian đáp ứng trung bình:  $(4 + 0 + 7 + 10 + 17)/5 = 7.6$

Thời gian hoàn thành trung bình:  $(28 + 14 + 12 + 31 + 23)/5 = 21.6$

Ví dụ 3

```
Enter algorithm (SJF, SRTF, RR, FCFS or EXIT): RR
Enter Quantum time: 5
```

Process Name	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time
1	0.00	13.00	31.00	31.00	18.00	0.00
2	8.00	7.00	33.00	25.00	18.00	9.00
3	3.00	6.00	23.00	20.00	14.00	2.00
4	6.00	2.00	17.00	11.00	9.00	9.00
5	11.00	16.00	44.00	33.00	17.00	12.00

Average waiting time : 15.20  
Average turn around time: 24.00  
Average response time: 6.40

Chạy tay

P1	P3	P1	P4	P2	P3	P5	P1	P2	P5	
0	5	10	15	17	22	23	28	31	33	44

Thời gian đáp ứng trung bình:  $(0 + 9 + 2 + 9 + 12)/5 = 6.4$

Thời gian đợi trung bình:  $((5 + 13) + (9 + 19) + (2 + 12) + 9 + (12 + 5))/5 = 15.2$

Thời gian hoàn thành trung bình:  $(31 + 25 + 20 + 11 + 33)/5 = 20.2$