

Group name : 4COS

ID: 21520497 – 21520472 – 21520213 - 21521924

Class: IT007.N11.1

OPERATING SYSTEM LAB 4'S REPORT

SUMMARY

Task		Status	Page
	Ex 1 : SJF	Hoàn thành	2
	Ex 2 : SRTF	Hoàn thành	11
	EX 3 : RR(Bonus)	Hoàn thành	21

Self-scores: 10

Note: Export file to **PDF and name the file by following format:
LAB4 – 2150497.pdf*

Code thực thi:

```
1  ///4COS
2  #include<iostream>
3  #include<math.h>
4
5  using namespace std;
6
7  struct Process{
8      string name;
9      int burst, arr;
10     int start, finish;
11     int rp, wt, tat;
12     bool completed;
13 };
14
15 int main(){
16     int n;
17     cout << "ENTER THE NUMBER OF PROCESS: "; cin >> n;
18     Process *p = new Process[n];
19     for(int i = 0; i < n; i++){
20         cout << "-----\n";
21         cout << "Enter the process name :";
22         cin >> p[i].name;
23         cout << "Enter arrival time :";
24         cin >> p[i].arr;
25         cout << "Enter burst time :";
26         cin >> p[i].burst;
27         p[i].completed = false;
28     }
29
30     int cur_t = 0, numProCompl = 0;
31     while(numProCompl != n){
32         int indBurstMin = -1;
33         int burstMin = INT_MAX;
34         for(int i = 0; i < n; i++){
35             if(p[i].arr <= cur_t && p[i].completed == false){
36                 if(p[i].burst < burstMin){
37                     indBurstMin = i;
38                     burstMin = p[i].burst;
39                 }
40                 else
41                 if(p[i].burst == burstMin)
42                 if(p[i].arr < p[indBurstMin].arr){
43                     indBurstMin = i;
44                     burstMin = p[i].burst;
45                 }
46             }
47         }
48         if(indBurstMin == -1)
49             cur_t += 1;
50         else{
51             p[indBurstMin].start = cur_t;
52             p[indBurstMin].finish = cur_t + p[indBurstMin].burst;
53             p[indBurstMin].rp = p[indBurstMin].start - p[indBurstMin].arr;
54             p[indBurstMin].wt = p[indBurstMin].rp;
55             p[indBurstMin].tat = p[indBurstMin].finish - p[indBurstMin].arr;
56             p[indBurstMin].completed = true;
57             numProCompl += 1;
58             cur_t = p[indBurstMin].finish;
59         }
60     }
```

```

61     float rpt_avg = 0;
62     float wt_avg = 0;
63     float tat_avg = 0;
64     for(int i = 0; i < n; i++){
65         rpt_avg += p[i].rp;
66         wt_avg += p[i].wt;
67         tat_avg += p[i].tat;
68     }
69     rpt_avg /= n;
70     wt_avg /= n;
71     tat_avg /= n;
72     cout << '\n';
73     for(int i = 0; i < n; i++){
74         cout << "Process Name: " << p[i].name << '\n';
75         cout << "\t\t- Waiting time: " << p[i].wt << '\n';
76         cout << "\t\t- Reponse time: " << p[i].rp << '\n';
77         cout << "\t\t- Turn Around time: " << p[i].tat << '\n';
78     }
79     cout << "-----SUMMARY-----\n";
80     cout << "- Average Waiting time: " << wt_avg << '\n';
81     cout << "- Average Reponse time: " << rpt_avg << '\n';
82     cout << "- Average Turn Around time: " << tat_avg << '\n';
83 }
84

```

Test File thực thi:

1. Test case 1:

Process Name	Arrival Time	Burst Time
P1	0	4
P2	5	1
P3	3	8
P4	10	2
P5	8	7

Bảng 1.1.1: Test case 1

Kết quả ước tính (bằng tay)

Process Name	Waiting Time	Response Time	Turnaround Time
P1	0	0	4
P2	7	7	8
P3	1	1	9
P4	3	3	5
P5	7	7	14
Average	3.6	3.6	8

Bảng 1.1.2: Kết quả ước tính (bằng tay) test case 1

Kết quả thực thi file Code chạy test case 1

```

minhtriet-21520497@minhtriet21520497-VirtualBox:~$ ls
bai2 bai2.c bai2.sh bai4 Desktop Documents Downloads ex3 Music Pictures Public sjf.cpp snap Templates Videos
minhtriet-21520497@minhtriet21520497-VirtualBox:~$ g++ -o sjf sjf.cpp
minhtriet-21520497@minhtriet21520497-VirtualBox:~$ ls
bai2 bai2.c bai2.sh bai4 Desktop Documents Downloads ex3 Music Pictures Public sjf sjf.cpp snap Templates Videos
minhtriet-21520497@minhtriet21520497-VirtualBox:~$ ./sjf
ENTER THE NUMBER OF PROCESS: 5
-----
Enter the process name :P1
Enter arrival time :0
Enter burst time :4
-----
Enter the process name :P2
Enter arrival time :5
Enter burst time :1
-----
Enter the process name :P3
Enter arrival time :3
Enter burst time :8
-----
Enter the process name :P4
Enter arrival time :10
Enter burst time :2
-----
Enter the process name :P5
Enter arrival time :8
Enter burst time :7
-----
Process Name: P1
- Waiting time: 0
- Reponse time: 0
- Turn Around time: 4
Process Name: P2
- Waiting time: 7
- Reponse time: 7
- Turn Around time: 8
Process Name: P3
- Waiting time: 1
- Reponse time: 1
- Turn Around time: 9
Process Name: P4
- Waiting time: 3
- Reponse time: 3
- Turn Around time: 5
Process Name: P5
- Waiting time: 7
- Reponse time: 7
- Turn Around time: 14
-----SUMMARY-----
- Average Waiting time: 3.6
- Average Reponse time: 3.6
- Average Turn Around time: 8

```

Hình 1.1: Kết quả thực thi code test case 1

2. Test case 2:

Process Name	Arrival Time	Burst Time
P1	0	8
P2	2	19
P3	4	3
P4	5	6
P5	7	12

Bảng 1.2.1: Test case 2

Kết quả ước tính (bằng tay)

Process Name	Waiting Time	Response Time	Turnaround Time
P1	0	0	8
P2	27	27	46
P3	4	4	7
P4	6	6	12
P5	10	10	22
Avegare	9,4	9,4	19

Bảng 1.2.2: Kết quả ước tính (bằng tay) test case 2

Kết quả thực thi Code test case 2:

```

minhtriet-21520497@minhtriet21520497-VirtualBox:~$ ./sjf
ENTER THE NUMBER OF PROCESS: 5
-----
Enter the process name :P1
Enter arrival time :0
Enter burst time :8
-----
Enter the process name :P2
Enter arrival time :2
Enter burst time :19
-----
Enter the process name :P3
Enter arrival time :4
Enter burst time :3
-----
Enter the process name :P4
Enter arrival time :5
Enter burst time :6
-----
Enter the process name :P5
Enter arrival time :7
Enter burst time :12
-----
Process Name: P1
- Waiting time: 0
- Reponse time: 0
- Turn Around time: 8
Process Name: P2
- Waiting time: 27
- Reponse time: 27
- Turn Around time: 46
Process Name: P3
- Waiting time: 4
- Reponse time: 4
- Turn Around time: 7
Process Name: P4
- Waiting time: 6
- Reponse time: 6
- Turn Around time: 12
Process Name: P5
- Waiting time: 10
- Reponse time: 10
- Turn Around time: 22
-----SUMMARY-----
- Average Waiting time: 9.4
- Average Reponse time: 9.4
- Average Turn Around time: 19
minhtriet-21520497@minhtriet21520497-VirtualBox:~$

```

Hình 1.2: Kết quả thực thi code test case 2

3. Test case 3:

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

Bảng 1.3.1: Test case 3

Kết quả ước tính (chạy bằng tay):

Process Name	Waiting Time	Response Time	Turnaround Time
P1	0	0	9
P2	5	5	10
P3	12	12	19
P4	13	13	23
P5	21	21	34
Average	10,2	10,2	19

Bảng 1.3.2: Kết quả ước tính (bằng tay) Test case 3

Kết quả thực thi Code test 3:

```

minhtriet-21520497@minhtriet21520497-VirtualBox:~$ ./sjf
ENTER THE NUMBER OF PROCESS: 5
-----
Enter the process name :P1
Enter arrival time :0
Enter burst time :9
-----
Enter the process name :P2
Enter arrival time :4
Enter burst time :5
-----
Enter the process name :P3
Enter arrival time :2
Enter burst time :7
-----
Enter the process name :P4
Enter arrival time :8
Enter burst time :10
-----
Enter the process name :P5
Enter arrival time :10
Enter burst time :13
-----
Process Name: P1
- Waiting time: 0
- Reponse time: 0
- Turn Around time: 9
Process Name: P2
- Waiting time: 5
- Reponse time: 5
- Turn Around time: 10
Process Name: P3
- Waiting time: 12
- Reponse time: 12
- Turn Around time: 19
Process Name: P4
- Waiting time: 13
- Reponse time: 13
- Turn Around time: 23
Process Name: P5
- Waiting time: 21
- Reponse time: 21
- Turn Around time: 34
-----SUMMARY-----
- Average Waiting time: 10.2
- Average Reponse time: 10.2
- Average Turn Around time: 19

```

Hình 1.3: Kết quả thực thi code test case 3

Chạy tay từng bước với test case 1

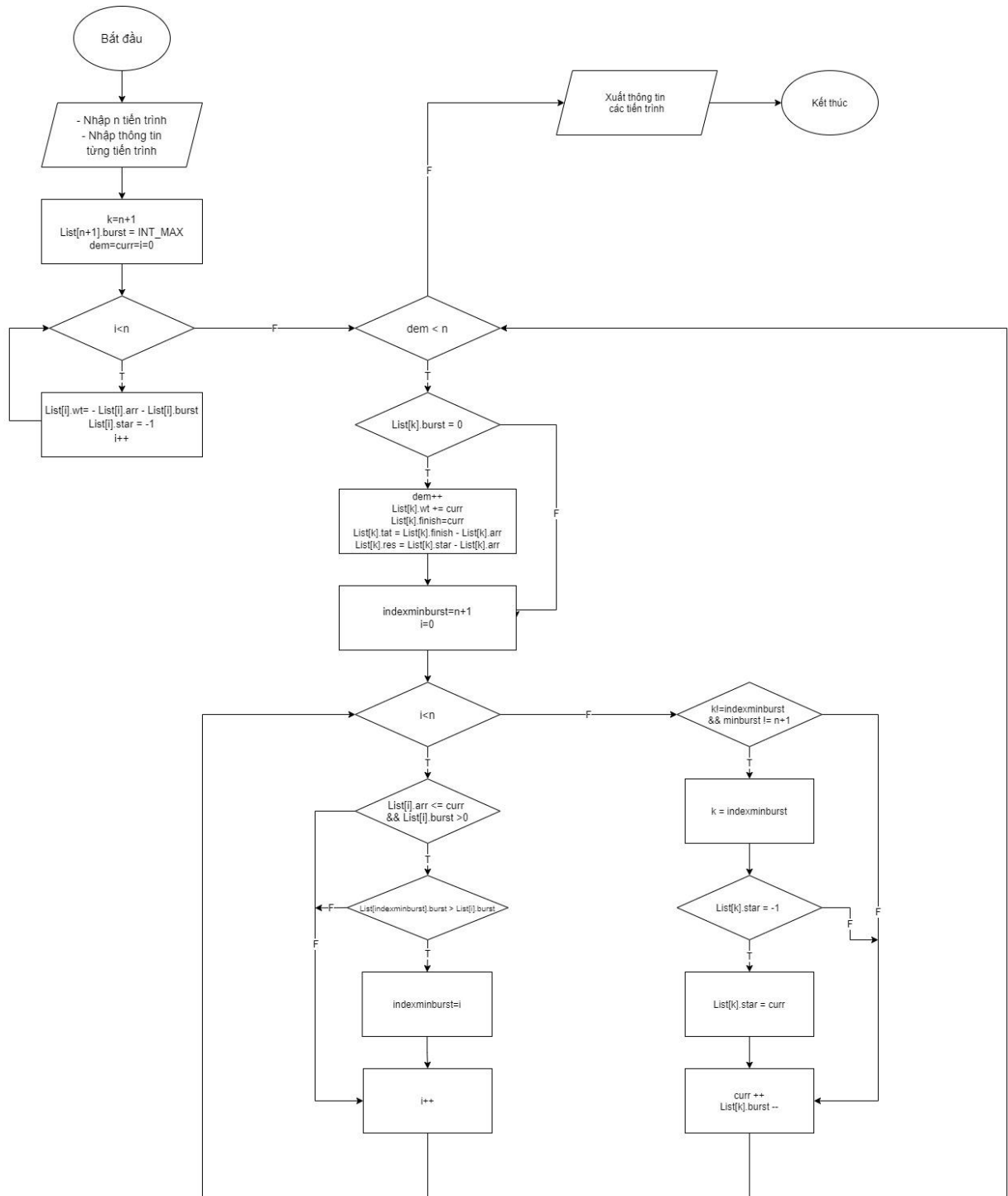
<pre>#include<iostream> #include<math.h> using namespace std; struct Process{ string name; int burst, arr; int start, finish; int rp, wt, tat; bool completed; }; int main(){ int n; cout << "ENTER THE NUMBER OF PROCESS: "; cin >> n; Process *p = new Process[n]; for(int i = 0; i < n; i++){ cout << "-----\n"; cout << "Enter the process name :"; cin >> p[i].name; cout << "Enter arrival time :"; cin >> p[i].arr; cout << "Enter burst time :"; cin >> p[i].burst; p[i].completed = false; } int cur_t = 0, numProCompl = 0;</pre>	<ul style="list-style-type: none"> - Bắt đầu - Nhập thông tin 5 tiến trình - cur_t = 0 - numProCompl = 0
<pre>while(numProCompl != n){ int indBurstMin = -1; int burstMin = INT_MAX; for(int i = 0; i < n; i++){ if(p[i].arr <= cur_t && p[i].completed == false){ if(p[i].burst < burstMin){ indBurstMin = i; burstMin = p[i].burst; } else if(p[i].burst == burstMin) if(p[i].arr < p[indBurstMin].arr){ indBurstMin = i; burstMin = p[i].burst; } } } numProCompl++; cur_t += burstMin; p[indBurstMin].start = cur_t - burstMin; p[indBurstMin].finish = cur_t; p[indBurstMin].rp = cur_t - p[indBurstMin].start; p[indBurstMin].wt = 0; p[indBurstMin].tat = p[indBurstMin].wt + p[indBurstMin].rp; p[indBurstMin].completed = true; }</pre>	<ul style="list-style-type: none"> - Ta có numProCompl != n (0 != 5) nên đi nhánh True - indBurstMin = -1; burstMin = INT_MAX; i = 0 - Với cur_t tìm tiến trình có burst time nhỏ nhất và tiến trình đó có arrival time <= cur_t - Với cur_t = 0 : <ul style="list-style-type: none"> + Ta có indBurstMin = 0 + Tiến hành cập nhật numProCompl = 1, cur_t = 4 + p[0].start = 0 + p[0].finish = 4 + p[0].rp = 0 + p[0].wt = 0 + p[0].tat = 4 + p[0].completed = true

<pre> if(indBurstMin == -1) cur_t += 1; else{ p[indBurstMin].start = cur_t; p[indBurstMin].finish = cur_t + p[indBurstMin].burst; p[indBurstMin].rp = p[indBurstMin].start - p[indBurstMin].arr; p[indBurstMin].wt = p[indBurstMin].rp; p[indBurstMin].tat = p[indBurstMin].finish - p[indBurstMin].arr; p[indBurstMin].completed = true; numProCompl += 1; cur_t = p[indBurstMin].finish; } } </pre>	<p>- Với cur_t = 4 :</p> <ul style="list-style-type: none"> + Ta có indBurstMin = 2 + Tiến hành cập nhật numProCompl = 2, cur_t = 12 + p[2].start = 4 + p[2].finish = 12 + p[2].rp = 1 + p[2].wt = 1 + p[2].tat = 9 + p[2].completed = true <p>- Với cur_t = 12 :</p> <ul style="list-style-type: none"> + Ta có indBurstMin = 1 + Tiến hành cập nhật numProCompl = 3, cur_t = 13 + p[1].start = 12 + p[1].finish = 13 + p[1].rp = 7 + p[1].wt = 7 + p[1].tat = 8 + p[1].completed = true <p>- Với cur_t = 13 :</p> <ul style="list-style-type: none"> + Ta có indBurstMin = 3 + Tiến hành cập nhật numProCompl = 4, cur_t = 15 + p[3].start = 13 + p[3].finish = 15 + p[3].rp = 3 + p[3].wt = 3 + p[3].tat = 5 + p[3].completed = true <p>- Với cur_t = 15 :</p> <ul style="list-style-type: none"> + Ta có indBurstMin = 4 + Tiến hành cập nhật numProCompl = 5, cur_t = 22 + p[4].start = 15 + p[4].finish = 22 + p[4].rp = 7 + p[4].wt = 7 + p[4].tat = 14 + p[4].completed = true
<pre> float rpt_avg = 0; float wt_avg = 0; float tat_avg = 0; for(int i = 0; i < n; i++){ rpt_avg += p[i].rp; wt_avg += p[i].wt; tat_avg += p[i].tat; } rpt_avg /= n; </pre>	<p>Áp dụng công thức để tính các thời gian và thời gian trung bình cần thiết, ta được kết quả:</p> <p>Process Name: P1</p> <ul style="list-style-type: none"> - Waiting time: 0 - Reponse time: 0 - Turn Around time: 4 <p>Process Name: P2</p>

<pre> wt_avg /= n; tat_avg /= n; cout << "\n"; for(int i = 0; i < n; i++){ cout << "Process Name: " << p[i].name << "\n"; cout << "\t\t- Waiting time: " << p[i].wt << "\n"; cout << "\t\t- Reponse time: " << p[i].rp << "\n"; cout << "\t\t- Turn Around time: " << p[i].tat << "\n"; } cout << "-----SUMMARY-----\n"; cout << "- Average Waiting time: " << wt_avg << "\n"; cout << "- Average Reponse time: " << rpt_avg << "\n"; cout << "- Average Turn Around time: " << tat_avg << "\n"; } </pre>	<pre> - Waiting time: 7 - Reponse time: 7 - Turn Around time: 8 Process Name: P3 - Waiting time: 1 - Reponse time: 1 - Turn Around time: 9 Process Name: P4 - Waiting time: 3 - Reponse time: 3 - Turn Around time: 5 Process Name: P5 - Waiting time: 7 - Reponse time: 7 - Turn Around time: 14 -----SUMMARY----- - Average Waiting time: 3.6 - Average Reponse time: 3.6 - Average Turn Around time: 8 </pre>
--	--

Câu 2: Vẽ lưu đồ, giải thích và hiện thực giải thuật SRTF

Lưu đồ:



Hình 2: Lưu đồ thuật toán SRTF

Giải thích:

- Đầu tiên, nhập thông tin và các thông số cần thiết cho List (Bao gồm tên tiến trình [name], arrival time [arr], burst time [burst])
- Thông tin các Process được cài đặt bao gồm: name (Process Name); arr (Arrival Time); burst (Burst Time); finish (Finish Time); star (Start Time); wt (Waiting Time); res (Response Time); tat (Turnaround Time)
- Khởi tạo biến $k=n+1$ với k là chỉ số của tiến trình đang chạy tại thời điểm
- Khởi tạo biến $dem = 0$ là biến đếm số tiến trình đã chạy hoàn tất
- Khởi tạo biến $curr$ là thời gian hiện hành
- Khởi tạo $List.star[i]=-1$ và $List.wt[i]=-List[i].arr-List[i].burst$ (Do $wt = finish - arr - burst$)
- Thực thi các tiến trình bằng cách:
 - + Nếu như tiến trình có $burst == 0$ (Tiến trình đã chạy xong) thì cập nhật các thông số của tiến trình
 - + Sử dụng vòng lặp để tìm ra chỉ số của tiến trình có burst time nhỏ nhất và arrival time \leq current time và chưa được chạy xong ($burst > 0$).
 - + Cập nhật lại $k = MinBurst$ nếu $MinBurst \neq k$ và nếu tiến trình được chạy lần đầu thì cập nhật start time cho tiến trình
 - + Trong quá trình chọn, tăng current time và giảm burst time của tiến trình đang chạy.
 - + Thực hiện liên tiếp cho đến khi đủ N Process được nhập vào thì tính trung bình của waiting time, response time, turnaround time và tiến hành xuất kết quả ra màn hình.

Code thực thi:

```
1  #include <iostream>
2  using namespace std;
3  struct process
4  {
5      string name;
6      int arr, burst, star, finish, tat, wt, res;
7  };
8
9  int main()
10 {
11     int n;
12     cout << "Enter number of process:";
13     cin >> n;
14     process List[100];
15     for (int i = 0; i < n; i++)
16     {
17         cout << "Enter name, arrival time, burst time of process " << i << " :";
18         cin >> List[i].name >> List[i].arr >> List[i].burst;
19     }
20
21     int k = n + 1;
22     List[n + 1].burst = INT16_MAX;
23     int Dem = 0;
24     int curr = 0;
25
26     for (int i = 0; i < n; i++)
27     {
28         List[i].wt = -List[i].arr - List[i].burst;
29         List[i].star = -1;
30     }
31 }
```

```

32 while (Dem < n)
33 {
34     if (List[k].burst == 0)
35     {
36         Dem++;
37         List[k].wt += curr;
38         List[k].finish = curr;
39         List[k].tat = List[k].finish - List[k].arr;
40         List[k].res = List[k].star - List[k].arr;
41     }
42     int MinBurst = n + 1;
43     for (int i = 0; i < n; i++)
44         if (List[i].arr <= curr && List[i].burst > 0)
45             if (List[MinBurst].burst > List[i].burst)
46                 MinBurst = i;
47     if (k != MinBurst && MinBurst != n + 1)
48     {
49         k = MinBurst;
50         if (List[k].star == -1) List[k].star = curr;
51     }
52     curr++;
53     List[k].burst--;
54 }
55 cout << "Name \t Start \t Finish \t Wt \t TAT \t res \n";
56 for (int i = 0; i < n; i++)
57     cout << List[i].name << '\t' << List[i].star << '\t' << List[i].finish << '\t' << List[i].wt << '\t' << List[i].tat<<
58 float aver_res=0, aver_tat=0, aver_wt=0;
59 for (int i = 0; i < n; i++)
60 {
61     aver_res += List[i].res;
62     aver_tat += List[i].tat;
63     aver_wt += List[i].wt;
64 }
65 aver_res /= n;
66 aver_tat /= n;
67 aver_wt /= n;
68 cout << "Average of Response Time: " << aver_res<<"\n";
69 cout << "Average of Turnaround Time: " << aver_tat<<"\n";
70 cout << "Average of waiting Time: " << aver_wt<<"\n";
71
72 }

```

Test File thực thi:

1. Test case 1:

Process Name	Arrival Time	Burst Time
P1	0	2
P2	4	20
P3	4	8
P4	9	3
P5	25	1
P6	12	2
P7	14	20

Bảng 2.1.1: Test case 1

Kết quả ước tính (bằng tay)

Process Name	Waiting Time	Response Time	Turnaround Time
P1	0	0	2
P2	14	13	34
P3	0	0	8
P4	5	5	8
P5	0	0	1
P6	0	0	2
P7	24	24	44
Average	6	14.1429	6.14286

Bảng 2.1.2: Kết quả ước tính (bằng tay) test case 1

Kết quả thực thi file Code chạy test case 1

```
thu-21520472@thu21520472-VirtualBox:~/srtf_lab4$ ./srtf
Enter number of process:7
Enter name, arrival time, burst time of process 0 :P1 0 2
Enter name, arrival time, burst time of process 1 :P2 4 20
Enter name, arrival time, burst time of process 2 :P3 4 8
Enter name, arrival time, burst time of process 3 :P4 9 3
Enter name, arrival time, burst time of process 4 :P5 25 1
Enter name, arrival time, burst time of process 5 :P6 12 2
Enter name, arrival time, burst time of process 6 :P7 14 20
Name    Start    Finish    Wt      TAT      res
P1       0         2         0        2         0
P2      17        38        14       34        13
P3       4        12         0         8         0
P4      14        17         5         8         5
P5      25        26         0         1         0
P6      12        14         0         2         0
P7      38        58        24       44        24
Average of Response Time: 6
Average of Turnaround Time: 14.1429
Average of waiting Time: 6.14286
thu-21520472@thu21520472-VirtualBox:~/srtf_lab4$
```

Hình 2.1: Kết quả thực thi code test case 1

2. Test case 2:

Process Name	Arrival Time	Burst Time
P1	0	8
P2	1	4
P3	2	2
P4	3	1
P5	4	3
P6	5	2

Bảng 1.2.1: Test case 2

Kết quả ước tính (bằng tay)

Process Name	Waiting Time	Response Time	Turnaround Time
P1	12	0	20
P2	5	0	9
P3	0	0	2
P4	1	1	2
P5	6	6	9
P6	0	0	2
Average	4	1.16667	7.33333

Bảng 1.2.2: Kết quả ước tính (bằng tay) test case 2

Kết quả thực thi Code test case 2:

```

thu-21520472@thu21520472-VirtualBox:~/srtf_lab4$ ./srtf
Enter number of process:6
Enter name, arrival time, burst time of process 0 :P1 0 8
Enter name, arrival time, burst time of process 1 :P2 1 4
Enter name, arrival time, burst time of process 2 :P3 2 2
Enter name, arrival time, burst time of process 3 :P4 3 1
Enter name, arrival time, burst time of process 4 :P5 4 3
Enter name, arrival time, burst time of process 5 :P6 5 2
Name  Start  Finish  Wt   TAT   res
P1    0      20      12   20    0
P2    1      10      5    9     0
P3    2       4      0    2     0
P4    4       5      1    2     1
P5   10     13      6    9     6
P6    5       7      0    2     0
Average of Response Time: 1.16667
Average of Turnaround Time: 7.33333
Average of waiting Time: 4
thu-21520472@thu21520472-VirtualBox:~/srtf_lab4$

```

Hình 2.2: Kết quả thực thi code test case 2

3. Test case 3:

Process Name	Arrival Time	Burst Time
P1	0	5
P2	3	1
P3	5	2
P4	8	7
P5	7	12

Bảng 2.3.1: Test case 3

Kết quả ước tính (chạy bằng tay):

Process Name	Waiting Time	Response Time	Turnaround Time
P1	1	0	6
P2	0	0	1
P3	1	1	3
P4	0	0	7
P5	8	8	20
Average	2	1.8	7.4

Bảng 2.3.2: Kết quả ước tính (bằng tay) Test case 3

Kết quả thực thi Code test 3:

```
thu-21520472@thu21520472-VirtualBox:~/srtf_lab4$ ./srtf
Enter number of process:5
Enter name, arrival time, burst time of process 0 :P1 0 5
Enter name, arrival time, burst time of process 1 :P2 3 1
Enter name, arrival time, burst time of process 2 :P3 5 2
Enter name, arrival time, burst time of process 3 :P4 8 7
Enter name, arrival time, burst time of process 4 :P5 7 12
Name    Start    Finish    Wt      TAT      res
P1       0        6         1        6         0
P2       3        4         0         1         0
P3       6        8         1         3         1
P4       8       15         0         7         0
P5      15       27         8        20         8
Average of Response Time: 1.8
Average of Turnaround Time: 7.4
Average of waiting Time: 2
thu-21520472@thu21520472-VirtualBox:~/srtf_lab4$
```

Hình 2.3: Kết quả thực thi code test case 3

Kết quả thực thi chạy tay:

Test case:

Process Name	Arrival Time	Burst Time
P1	0	4
P2	3	2
P3	6	7
P4	10	5
P5	15	3

Quá trình thực thi:

<pre>#include <iostream> using namespace std; struct process { string name; int arr, burst, star, finish, tat, wt, res; }; int main() { int n; cout << "Enter number of process:"; cin >> n; process List[100]; for (int i = 0; i < n; i++) { cout << "Enter name, arrival time, burst time of process " << i << " :"; cin >> List[i].name >> List[i].arr >> List[i].burst; } int k = n + 1; List[n + 1].burst = INT16_MAX; int Dem = 0; int curr = 0; for (int i = 0; i < n; i++) { List[i].wt = -List[i].arr - List[i].burst; List[i].star = -1; } }</pre>	<p>Khai báo và nhập các thông tin cần thiết</p> <p>n=5 k=6 Dem=0 Curr=0</p>
<pre>while (Dem < n) { if (List[k].burst == 0) { Dem++; List[k].wt += curr; List[k].finish = curr; List[k].tat = List[k].finish - List[k].arr; List[k].res = List[k].star - List[k].arr; } int MinBurst = n + 1;</pre>	<p>Với curr = 0 Minburst:0 k:0 List[0].star:0 curr:1; List[0].burst: 3</p> <p>Với curr = 1 Minburst:0 curr:2; List[0].burst: 2</p> <p>Với curr = 2</p>

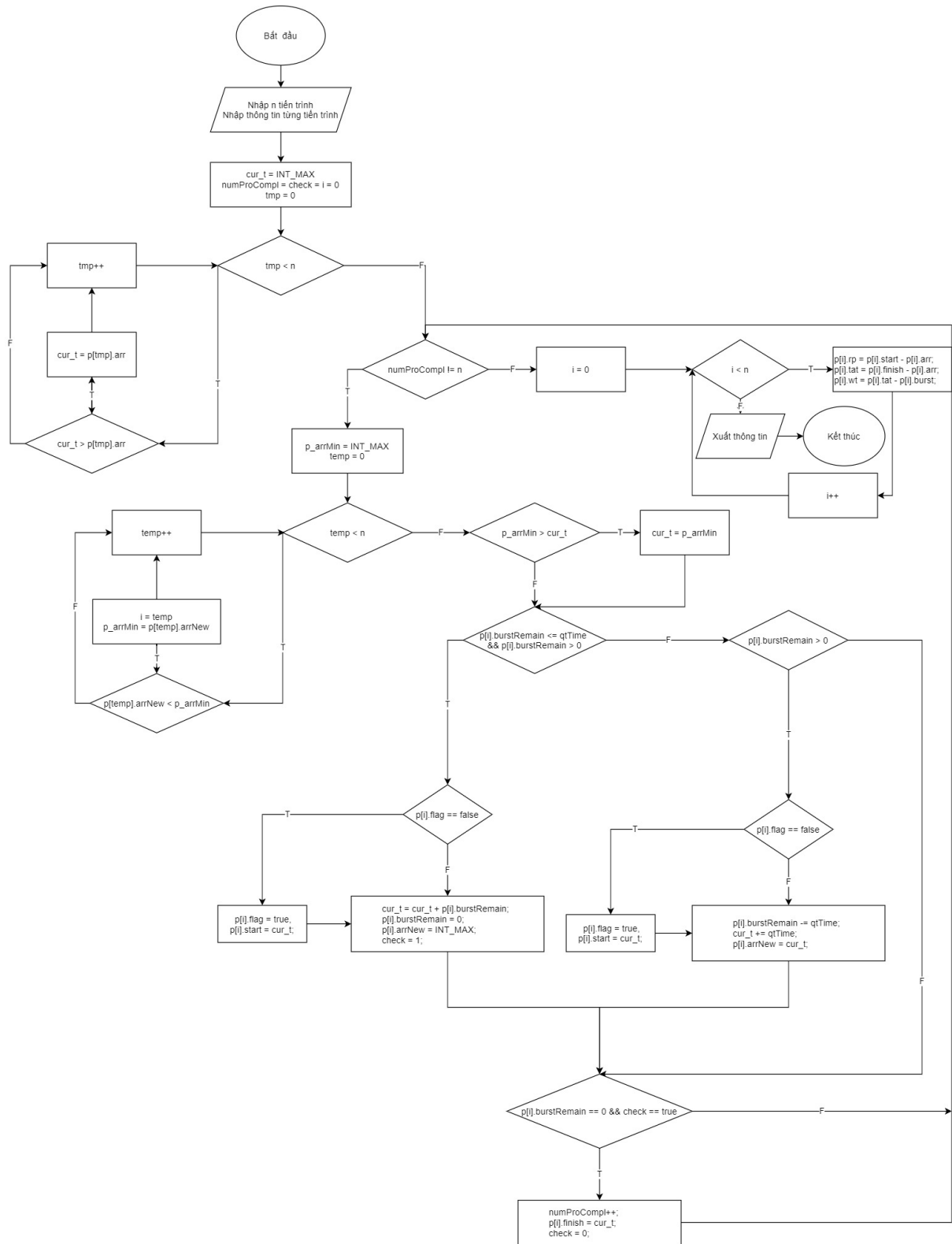
<pre> for (int i = 0; i < n; i++) if (List[i].arr <= curr && List[i].burst > 0) if (List[MinBurst].burst > List[i].burst) MinBurst = i; if (k != MinBurst && MinBurst != n + 1) { k = MinBurst; if (List[k].star == -1) List[k].star = curr; } curr++; List[k].burst--; } </pre>	<p>Minburst:0 curr:3 List[0].burst: 1</p> <p>Với curr = 3 Minburst:0 curr:4 List[0].burst: 0</p> <p>Với curr = 4 Dem:1 Cập nhật thông số cho tiến trình: wt:0 finish: 4 tat : 4 res : 0 Minburst:1 k:1 List[1].star:4 curr:5; List[1].burst: 1</p> <p>Với curr = 5 Minburst:1 curr:6; List[k].burst: 0</p> <p>Với curr = 6 Dem:2 Cập nhật thông số cho tiến trình: wt:1; finish: 6; tat : 3; res : 1 Minburst:2 k:2 List[2].star:6 curr:7 List[2].burst: 6</p> <p>Với curr = 7 Minburst:2 curr:8 List[2].burst: 5</p> <p>Với curr = 8 Minburst:2 curr:9 List[2].burst: 4</p> <p>Với curr = 9 Minburst:2 curr:10 List[2].burst: 3</p> <p>Với curr = 10 Minburst:2 curr:11 List[2].burst: 2</p> <p>Với curr = 11 Minburst:2 curr:12 List[2].burst: 1</p> <p>Với curr = 12 Minburst:2 curr:13 List[2].burst: 0</p> <p>Với curr = 13</p>
--	--

	<p>Dem:3 Cập nhật thông số cho tiến trình: wt:0 finish: 13 tat : 7 res : 0 Minburst:3 k:3 List[3].star:13 curr:14 List[3].burst: 4</p> <p>Với curr = 14 Minburst:3 curr:15 List[3].burst: 3</p> <p>Với curr = 15 Minburst:3 curr:16 List[3].burst: 2</p> <p>Với curr = 16 Minburst:3 curr:17 List[k].burst: 1</p> <p>Với curr = 17 Minburst:3 curr:18 List[k].burst: 0</p> <p>Với curr = 18 Dem:4 Cập nhật thông số cho tiến trình: wt:3 finish: 18 tat : 8 res : 3 Minburst:4 k:4 List[k].star:18 curr:19 List[k].burst: 2</p> <p>Với curr = 19 Minburst:4 curr:20 List[k].burst: 1</p> <p>Với curr = 20 Minburst:4 curr:21 List[k].burst: 0</p> <p>Với curr = 21 Dem:5 Cập nhật thông số cho tiến trình: wt:3 finish: 21 tat : 6 res : 3 Minburst:6 curr:22 List[k].burst: -1</p>
--	---

<pre> cout << "Name \t Start \t Finish \t Wt \t TAT \t res \n"; for (int i = 0; i < n; i++) cout << List[i].name << '\t' << List[i].star << '\t' << List[i].finish << '\t' << List[i].wt << '\t' << List[i].tat<<'\t'<<List[i].res << '\n'; float aver_res=0, aver_tat=0, aver_wt=0; for (int i = 0; i < n; i++) { aver_res += List[i].res; aver_tat += List[i].tat; aver_wt += List[i].wt; } aver_res /= n; aver_tat /= n; aver_wt /= n; cout << "Average of Response Time: " << aver_res<<"\n"; cout << "Average of Turnaround Time: " << aver_tat<<"\n"; cout << "Average of waiting Time: " << aver_wt<<"\n"; </pre>	<p>Xuất thông tin các tiến trình và tính trung bình thời gian cần thiết, ta được kết quả như sau:</p> <p>Average of Response Time: 1.4 Average of Turnaround Time: 5.6 Average of waiting Time: 1.4</p>
--	---

Câu 3: Vẽ lưu đồ, giải thích và hiện thực giải thuật RR

Lưu đồ:



Hình 3: Lưu đồ thuật toán RR

Giải thích:

- Nhập thông tin cho danh sách n tiến trình gồm : name(Process Name), arr(Arrival Time), burst(Burst Time). qtTime(Quantum Time)
- Thông tin của các tiến trình cần tính toán gồm có : name(Process Name), arr(Arrival Time), burst(Burst Time), start(Start Time), finish(Finish Time), wt(Waiting Time), rp(Response Time), tat(Turn Around Time), flag (Cho biết tiến trình đã thực thi lần nào hay chưa true – đã thực thi, false – chưa thực thi), burstRemain(Burst time cần phải thực thi).
- Khởi tạo cur_t = 0 (cur_t : thời điểm hiện hành), numProCompl = 0 (Số tiến trình đã thực thi hoàn thành).
- Chọn cur_t có giá trị của tiến trình có arrival time là bé nhất để chọn làm mốc bắt đầu.
- Chương trình thực thi các tiến trình bằng thuật toán RR như sau :
 - o Dùng vòng lặp chọn Process có Arrival Time New nhỏ nhất để thực thi (cập nhật lại giá trị Current Time sau mỗi lần thực thi), trong quá trình chọn, nếu thời điểm hiện tại nhỏ hơn Arrival Time New của tiến trình có Arrival Time New bé nhất thì cập nhật lại cur_t
 - o Tiến hành xét tiến trình thứ i nếu có burstRemain <= qtTime và burstRemain > 0 thì cập nhật lại những thông tin cần thiết(numProCompl++, p[i].finish = cur_t, check = 0).
 - o Nếu tiến trình đó chưa thể hoàn thành trong quantum time lần này thì tiến hành cho thực thi và cập nhật lại thông tin cần thiết(cur_t += qtTime, p[i].burstRemain -= qtTime, p[i].arrNew = cur_t).
 - o Thực hiện liên tiếp cho đến khi đủ N Process được nhập vào thì tiến hành xuất kết quả ra màn hình.

Cocde thực thi:

```
1  #include<iostream>
2  #include<math.h>
3  #include<algorithm>
4  using namespace std;
5
6  struct Process{
7      string name;
8      int burst, arr;
9      int start, finish;
10     int arrNew;
11     int rp, wt, tat;
12     bool flag;
13     int burstRemain;
14 };
15
16 bool cmp(Process a, Process b){
17     return a.arr < b.arr;
18 }
19
20 int main(){
21     int n, qtTime;
22     cout << "ENTER QUANTUM TIME: "; cin >> qtTime;
23     cout << "ENTER THE NUMBER OF PROCESS: "; cin >> n;
24     Process *p = new Process[n];
25     for(int i = 0; i < n; i++){
26         cout << "-----\n";
27         cout << "Enter the process name :"; cin >> p[i].name;
28         cout << "Enter arrival time :"; cin >> p[i].arr;
29         cout << "Enter burst time :"; cin >> p[i].burst;
30         p[i].burstRemain = p[i].burst;
31         p[i].arrNew = p[i].arr;
32         p[i].flag = false;
33     }
34
35     int cur_t = INT_MAX, i = 0, numProCompl = 0, check = 0;
36 }
```

```

37     for(int tmp = 0; tmp < n; tmp++)
38         if(cur_t > p[tmp].arr)
39             cur_t = p[tmp].arr;
40
41     while(numProCompl != n){
42         int p_arrMin = INT_MAX;
43
44         for(int tmp = 0; tmp < n; tmp++)
45             if(p[tmp].arrNew < p_arrMin)
46                 i = tmp, p_arrMin = p[tmp].arrNew;
47         if(p_arrMin > cur_t)
48             cur_t = p_arrMin;
49         if(p[i].burstRemain <= qtTime && p[i].burstRemain > 0)
50         {
51             if(!p[i].flag)
52                 p[i].flag = true, p[i].start = cur_t;
53             cur_t = cur_t + p[i].burstRemain;
54             p[i].burstRemain = 0;
55             p[i].arrNew = INT_MAX;
56             check = 1;
57         }
58         else
59         if(p[i].burstRemain > 0){
60             if(!p[i].flag)
61                 p[i].flag = true, p[i].start = cur_t;
62             p[i].burstRemain -= qtTime;
63             cur_t += qtTime;
64             p[i].arrNew = cur_t;
65         }
66
67         if(p[i].burstRemain == 0 && check){
68             numProCompl++;
69             p[i].finish = cur_t;
70             check = 0;
71         }
72     }
73
74     for(int i = 0; i < n; i++)
75     {
76         p[i].rp = p[i].start - p[i].arr;
77         p[i].tat = p[i].finish - p[i].arr;
78         p[i].wt = p[i].tat - p[i].burst;
79     }
80
81     float rpt_avg = 0;
82     float wt_avg = 0;
83     float tat_avg = 0;
84     for(int i = 0; i < n; i++){
85         rpt_avg += p[i].rp;
86         wt_avg += p[i].wt;
87         tat_avg += p[i].tat;
88     }
89     rpt_avg /= n;
90     wt_avg /= n;
91     tat_avg /= n;
92     cout << '\n';
93     for(int i = 0; i < n; i++){
94         cout << "Process Name: " << p[i].name << '\n';
95         cout << "\t\t- Waiting time: " << p[i].wt << '\n';
96         cout << "\t\t- Reponse time: " << p[i].rp << '\n';
97         cout << "\t\t- Turn Around time: " << p[i].tat << '\n';
98     }
99     cout << "-----SUMMARY-----\n";
100    cout << "- Average Waiting time: " << wt_avg << '\n';
101    cout << "- Average Reponse time: " << rpt_avg << '\n';
102    cout << "- Average Turn Around time: " << tat_avg << '\n';
103 }
104

```

Test File thực thi:

1. Test case 1: Quantum time = 5

Process Name	Arrival Time	Burst Time
P1	0	13
P2	4	9
P3	6	4
P4	7	20
P5	12	10

Bảng 1.1.1: Test case 1

Kết quả ước tính (bằng tay)

Process Name	Waiting Time	Response Time	Turnaround Time
P1	23	0	36
P2	15	1	24
P3	9	9	13
P4	29	12	49
P5	24	16	34
Average	20	7.6	31.2

Bảng 1.1.2: Kết quả ước tính (bằng tay) test case 1

Kết quả thực thi file Code chạy test case 1

```
phucdanh-21521924@phucdanh21521924-virtual-machine:~/Downloads$ ./rr
ENTER QUANTUM TIME: 5
ENTER THE NUMBER OF PROCESS: 5
-----
Enter the process name :1
Enter arrival time :0
Enter burst time :13
-----
Enter the process name :2
Enter arrival time :4
Enter burst time :9
-----
Enter the process name :3
Enter arrival time :6
Enter burst time :4
-----
Enter the process name :4
Enter arrival time :7
Enter burst time :20
-----
Enter the process name :5
Enter arrival time :12
Enter burst time :10

Process Name: 1
- Waiting time: 23
- Reponse time: 0
- Turn Around time: 36
Process Name: 2
- Waiting time: 15
- Reponse time: 1
- Turn Around time: 24
Process Name: 3
- Waiting time: 9
- Reponse time: 9
- Turn Around time: 13
Process Name: 4
- Waiting time: 29
- Reponse time: 12
- Turn Around time: 49
Process Name: 5
- Waiting time: 24
- Reponse time: 16
- Turn Around time: 34
-----SUMMARY-----
- Average Waiting time: 20
- Average Reponse time: 7.6
- Average Turn Around time: 31.2
phucdanh-21521924@phucdanh21521924-virtual-machine:~/Downloads$
```

Hình 3.1: Kết quả thực thi code test case 1

2. Test case 2: Quantum time = 5

Process Name	Arrival Time	Burst Time
P1	0	13
P2	8	7
P3	3	6
P4	6	2
P5	11	16

Bảng 1.2.1: Test case 2

Kết quả ước tính (bằng tay)

Process Name	Waiting Time	Response Time	Turnaround Time
P1	18	0	31
P2	18	9	25
P3	14	2	20
P4	9	9	11
P5	17	12	33
Average	15.2	6.4	24

Bảng 1.2.2: Kết quả ước tính (bằng tay) test case 2

Kết quả thực thi Code test case 2:

```
phucdanh-21521924@phucdanh21521924-virtual-machine:~/Downloads$ ./rr
ENTER QUANTUM TIME: 5
ENTER THE NUMBER OF PROCESS: 5
-----
Enter the process name :1
Enter arrival time :0 13
Enter burst time :-----
Enter the process name :2
Enter arrival time :8
Enter burst time :7
-----
Enter the process name :3
Enter arrival time :3
Enter burst time :6
-----
Enter the process name :4
Enter arrival time :6
Enter burst time :2
-----
Enter the process name :5
Enter arrival time :11
Enter burst time :16
-----
Process Name: 1
- Waiting time: 18
- Reponse time: 0
- Turn Around time: 31
Process Name: 2
- Waiting time: 18
- Reponse time: 9
- Turn Around time: 25
Process Name: 3
- Waiting time: 14
- Reponse time: 2
- Turn Around time: 20
Process Name: 4
- Waiting time: 9
- Reponse time: 9
- Turn Around time: 11
Process Name: 5
- Waiting time: 17
- Reponse time: 12
- Turn Around time: 33
-----SUMMARY-----
- Average Waiting time: 15.2
- Average Reponse time: 6.4
- Average Turn Around time: 24
phucdanh-21521924@phucdanh21521924-virtual-machine:~/Downloads$ S
```

Hình 3.2: Kết quả thực thi code test case 3

3. Test case 3: Quantum Time = 10

Process Name	Arrival Time	Burst Time
P1	0	10
P2	4	29
P3	8	3
P4	38	7
P5	40	12

Bảng 1.3.1: Test case 3

Kết quả ước tính (chạy bằng tay):

Process Name	Waiting Time	Response Time	Turnaround Time
P1	0	0	10
P2	9	6	38
P3	12	12	15
P4	4	4	11
P5	9	9	21
Average	6.8	6.2	19

Bảng 1.3.2: Kết quả ước tính (bằng tay) Test case 3

Kết quả thực thi Code test 3:

```
phucdanh-21521924@phucdanh21521924-virtual-machine:~/Downloads$ ./rr
ENTER QUANTUM TIME: 10
ENTER THE NUMBER OF PROCESS: 5
-----
Enter the process name :1
Enter arrival time :0
Enter burst time :10
-----
Enter the process name :2
Enter arrival time :4
Enter burst time :29
-----
Enter the process name :3
Enter arrival time :8
Enter burst time :3
-----
Enter the process name :4
Enter arrival time :38
Enter burst time :7
-----
Enter the process name :5
Enter arrival time :40
Enter burst time :12
-----
Process Name: 1
- Waiting time: 0
- Reponse time: 0
- Turn Around time: 10
Process Name: 2
- Waiting time: 9
- Reponse time: 6
- Turn Around time: 38
Process Name: 3
- Waiting time: 12
- Reponse time: 12
- Turn Around time: 15
Process Name: 4
- Waiting time: 4
- Reponse time: 4
- Turn Around time: 11
Process Name: 5
- Waiting time: 9
- Reponse time: 9
- Turn Around time: 21
-----SUMMARY-----
- Average Waiting time: 6.8
- Average Reponse time: 6.2
- Average Turn Around time: 19
phucdanh-21521924@phucdanh21521924-virtual-machine:~/Downloads$
```

Hình 3.3: Kết quả thực thi code test case 3

Kết quả thực thi bằng tay:

Test case: Quantum time = 5

Process Name	Arrival Time	Burst Time
P1	0	13
P2	4	9
P3	6	4
P4	7	20
P5	12	10

Quá trình thực thi:

```
#include<iostream>
#include<math.h>
#include<algorithm>
using namespace std;

struct Process {
    string name;
    int burst, arr;
    int start, finish;
    int arrNew;
    int rp, wt, tat;
    bool flag;
    int burstRemain;
};

bool cmp(Process a, Process b) {
    return a.arr < b.arr;
}

int main() {
    int n, qtTime;
    cout << "ENTER QUANTUM TIME: "; cin >>
    qtTime;
    cout << "ENTER THE NUMBER OF PROCESS: ";
    cin >> n;
    Process* p = new Process[n];
    for (int i = 0; i < n; i++) {
        cout << "-----\n";
        cout << "Enter the process name :"; cin >>
        p[i].name;
        cout << "Enter arrival time :"; cin >> p[i].arr;
        cout << "Enter burst time :"; cin >> p[i].burst;
        p[i].burstRemain = p[i].burst;
        p[i].arrNew = p[i].arr;
        p[i].flag = false;
    }

    int cur_t = INT_MAX, i = 0, numProCompl = 0, check
    = 0;

    for (int tmp = 0; tmp < n; tmp++)
        if (cur_t > p[tmp].arr)
            cur_t = p[tmp].arr;
```

Khai báo và nhập các thông tin cần thiết

Ta có:

cur_t = INT_MAX,

i = 0, numProCompl = 0,

check = 0

cur_t = 0

<pre> while (numProCompl != n) { int p_arrMin = INT_MAX; for (int tmp = 0; tmp < n; tmp++) if (p[tmp].arrNew < p_arrMin) i = tmp, p_arrMin = p[tmp].arrNew; if (p_arrMin > cur_t) cur_t = p_arrMin; if (p[i].burstRemain <= qtTime && p[i].burstRemain > 0) { if (!p[i].flag) p[i].flag = true, p[i].start = cur_t; cur_t = cur_t + p[i].burstRemain; p[i].burstRemain = 0; p[i].arrNew = INT_MAX; check = 1; } else if (p[i].burstRemain > 0) { if (!p[i].flag) p[i].flag = true, p[i].start = cur_t; p[i].burstRemain -= qtTime; cur_t += qtTime; p[i].arrNew = cur_t; } if (p[i].burstRemain == 0 && check) { numProCompl++; p[i].finish = cur_t; check = 0; } } </pre>	<pre> p_arrMin=0 Ta có: p[0].burstRemain > qtTime && p[0].BurstRemain > 0 nên: p[0].flag = true và p[0].start =0 p[0].burstRemain=8 p[0].arrNew = 5 cur_t=5 p_arrMin=4 Ta có:p[1].burstRemain > qtTime && p[1].BurstRemain > 0 nên: p[1].flag = true và p[1].start =5 p[1].burstRemain=4 p[1].arrNew = 10 cur_t =10 p_arrMin=5 Ta có:p[0].burstRemain > qtTime && p[0].BurstRemain > 0 nên: p[0].burstRemain=3 p[0].arrNew = 15 cur_t=15 p_arrMin=6 Ta có: p[2].burstRemain <= qtTime && p[2].burstRemain > 0 nên: p[2].flag = true và p[2].start =15 cur_t=19 p[2].burstRemain=0 p[2].arrNew = INT_MAX Check = 1 Tiến trình p[2] đã được hoàn thành nên: numProCompl=1; p[2].finish=19 p_arrMin=7 Ta có:p[3].burstRemain > qtTime && p[3].BurstRemain > 0 nên: p[3].flag = true và p[3].start =19 p[3].burstRemain=15 p[3].arrNew = 24 cur_t=24 p_arrMin=10 Ta có: p[1].burstRemain <= qtTime && p[1].burstRemain > 0 nên: p[1].flag = true và p[1].start =24 cur_t=28 p[1].burstRemain=0 p[1].arrNew = INT_MAX Check = 1 Tiến trình p[1] đã được hoàn thành nên: numProCompl=2; p[1].finish=28 p_arrMin=12 </pre>
--	---

	<p>Ta có: p[4].burstRemain > qtTime && p[4].BurstRemain > 0 nên: p[4].flag = true và p[4].start = 28 p[4].burstRemain = 5 p[4].arrNew = 33 cur_t = 33</p> <p>p_arrMin = 15 Ta có: p[0].burstRemain <= qtTime && p[0].burstRemain > 0 nên: p[0].flag = true và p[0].start = 33 cur_t = 36 p[0].burstRemain = 0 p[0].arrNew = INT_MAX Check = 1 Tiến trình p[0] đã được hoàn thành nên: numProCompl = 3; p[0].finish = 36</p> <p>p_arrMin = 24 Ta có: p[3].burstRemain > qtTime && p[3].BurstRemain > 0 nên: p[3].burstRemain = 10 p[3].arrNew = 41 cur_t = 41</p> <p>p_arrMin = 33 Ta có: p[4].burstRemain <= qtTime && p[4].burstRemain > 0 nên: p[4].flag = true và p[4].start = 41 cur_t = 46 p[4].burstRemain = 0 p[4].arrNew = INT_MAX Check = 1 Tiến trình p[4] đã được hoàn thành nên: numProCompl = 4; p[4].finish = 46</p> <p>p_arrMin = 41 Ta có: p[3].burstRemain > qtTime && p[3].BurstRemain > 0 nên: p[3].burstRemain = 5 p[3].arrNew = 51 cur_t = 51</p>
<pre> for (int i = 0; i < n; i++) { p[i].rp = p[i].start - p[i].arr; p[i].tat = p[i].finish - p[i].arr; p[i].wt = p[i].tat - p[i].burst; } float rpt_avg = 0; float wt_avg = 0; float tat_avg = 0; for (int i = 0; i < n; i++) { rpt_avg += p[i].rp; wt_avg += p[i].wt; tat_avg += p[i].tat; } </pre>	<p>Áp dụng công thức để tính các thời gian và thời gian trung bình cần thiết, ta được kết quả:</p> <p>Process Name: P1 - Waiting time: 23 - Reponse time: 0 - Turn Around time: 36</p> <p>Process Name: P2 - Waiting time: 15 - Reponse time: 1 - Turn Around time: 24</p> <p>Process Name: P3</p>

```

}
rpt_avg /= n;
wt_avg /= n;
tat_avg /= n;
cout << '\n';
for (int i = 0; i < n; i++) {
    cout << "Process Name: " << p[i].name << '\n';
    cout << "\t\t- Waiting time: " << p[i].wt << '\n';
    cout << "\t\t- Reponse time: " << p[i].rp << '\n';
    cout << "\t\t- Turn Around time: " << p[i].tat << '\n';
}
cout << "-----SUMMARY-----\n";
cout << "- Average Waiting time: " << wt_avg << '\n';
cout << "- Average Reponse time: " << rpt_avg << '\n';
cout << "- Average Turn Around time: " << tat_avg <<
'\n';

```

```

- Waiting time: 9
- Reponse time: 9
- Turn Around time: 13
Process Name: P4
- Waiting time: 29
- Reponse time: 12
- Turn Around time: 49
Process Name: P5
- Waiting time: 24
- Reponse time: 16
- Turn Around time: 34
-----SUMMARY-----
- Average Waiting time: 20
- Average Reponse time: 7.6
- Average Turn Around time: 31.2

```

.....HẾT.....