Write a C-program to simulate the CPU scheduling algorithms first come First serve (FCFS) and shortest job first (Non-preemption).

```
#include <stdio.h>
int at[20], cput[20];
void main() {
int n, i, choice;
Printf("Enter number of processes \n");
Scanf("%d", &n);
Printf("Enter arrival time & CPU time for each process respectively\n");
for (i=0; i<n; i++) {
scanf("%d%d", & at[i], & cput[i]);
}
Printf("Menue\n\n 1.FCFS \n2. SJF(Non-preemtive)
\n 3. SRTF (Preemptive) \n 4. Exit\n);
while (1) {
Scanf("%d", &choice);
switch (choice) {
case1: FCFS(n);
break;
case 2: SJF(n);
break;
case 3: SRTF(n);
break;
case 4: exit(0);
default: printf("Wrong choice \n");
}
}
}
void FCFS(int n) {
int cmpt[20], tat[20], wt[20], pname[20], temp;
float awt =0, atat=0;
int sum=0, i;
for(i=0; i<n;i++) {
pname[i]=i;
}
for (i=0; i<n;i++) {
if (at[i] = = at[i+1] && cput[i]>cput[i+1])
 temp=cput[i];
  cput[i]=cput[i+1];
 cput[i+1]=temp;
  temp = pname [i];
 pname [i] = pname [i+1];
  pname [i+1] = temp;
 }
  }
 for (i=0; i<n;i++) {
 sum+=cput[i];
  cmpt[i]=sum;
  tat[i]=cmpt[i]-at[i];
  wt[i] = tat[i]-cput[i];
  }
  for(i=0; i<n; i++) {
  awt += wt[i];
 atat += tat[i];
   }
  awt= awt/n;
  atat = atat /n;
 Printf ("\t Process\t Arrival time \t cputime \t
 waitingtime \t turn around time \n");
 for (i=0; i<n;i++){
 Printf ("\n\t p%d\t \t %d\t\t %d \t\t
 %d", pname[i], at [i], cput [i], wt [i], at[i]);
  }
 Printf ("Average waiting time --. %d", awt);
 Prinr f ("\n Average turnaround time --- %f", atat");
 }
```

```
void sJF (intn){
int cmpt [20], tat [20], wt [20], cput1 [20];
float awt=0, atat=0, sum brust- time =0;
int sum=0, i, j, smallest;
for (i=0; i<n; i++) { cput1[i] = cput [i]
Sum_burst_time += cput [i]; }
Printf ("\t Process \t waiting time \t turnaround
time \n");
cput1 [9] = 9999;
while (sum < sum_burst_time){
  smallest = 9;
  for(i=0; i<n; i++){
  if (at[i] <= sum && cput [i] > 0 && cput1[i]
  < cput1 [smallest])
      smallest=i;
  }
  Printf ("\t P[%d] \t\t %d\t \t %d\n",
  smallest, sum + cput1 [smallest] - at [smallest],
     sum - at [smallest]);
  awt += sum + cput [smallest] - at [smallest];
  atat += sum -at [smallest];
  cput1 [smallest]=0;
  }
awt = awt /n;
atat = atat /n;
Printf ("\n Arerage wait time -- %f ", awt);
Printf ("\n Arerage turnaround tine --- %f\n", atat);
}
void srtf (intn){
int remaining_time [20], tat [20], wt [20],
completion time [w], smallest, time, i, count=0;
float awt=0, atat=0;
for (i=0; i<n; i++)
remaining_time [i] = cput [i];
```

```
time = 0;
while (count != n) {
smallest = -1;
for (i=0; i<n; i++) {
 if (at + i) < = time && remaining-time [i] > 0
{
 if (smallest == -1 || remaining-time[i] < remaining-
time [smallest])
   smallest = i;
   }
 }
 if (smallest = = -1) {
   time ++;
   continue;
 }
 remaining-time [smallest] - -;
 if (remaining-time [smallest] = = 0) {
  count ++;
completion-time [smallest ] = time + 1;
wt [smallest = completion-time [smallent] -
 at [smallest] - cput [smallest];
  tat [smallest] = completion-time [smallest] -
         at [smallest];
  }
time + +;
}
for (i=0; i<n; i++) {
 awt + = wt [i];
 atat + = tat [i];
 }
awt = awt/n;
atat = atat/n;
Printf ("/n Process \n Avial ~~timat~~ time \n CPU time \n
waiting time \n Turn around time \n");
for(i=0;i<n;i++){
Printf("%d\t %d\t %d\t%d\t; i",
at[i], cput[i], wt[i], tat[i]);
}
Printf("Average waiting time--- %f", awt);
Printf("Average turnaround time --% f \n", atat);
}
```

Output

```
Enter number of process
4
Enter arrival time and CPU time for each process respectively
 0  3
 1  6
 4  4
 6  2
Menue
1. FCFS
2. STF
3. SRTF
4. Exit
1.
Process  Arrival  CPU  Waiting  turn around
 P0        0       3     0         3
 P1        1       4     2         8
 P2        4       6     5         9
 P3        6       2     7         9
Average waiting time -- 3.50000
Average turn around time -- 7.2500
2
Process   waiting time   Turn arount
P[0]          3              0
P[1]          8              2
P[3]          5              3
P[2]          11             7
```

Average waiting time -- 6.75000
Average turnaround time -- 3.0000

3

Process	Arrival time	CPU	waiting	Turnaround
0	0	3	0	3
1	1	6	8	14
2	4	4	0	4
3	6	2	2	4

Averag waiting time -- 2.50000.
Average Turn around time -6.250000

1. FCFS

P0	P0	P2	P 1	P1	P2	P3	
0	1	3	4	6	9	13	15

P(0)3	P0(2)
	P1(6)
		P1(6)	P1(5)
			P2(4)
				P1(3)
				P2(4)
				P3(2)
					P2(4)
					P3(2)
						P3(2)	

2. SJF

P0	P0	P1	P1	P1	P3	P2	
0	1	3	4	6	9	11	15

P0(3)	P0(2)
	P1(6)
		P1(6)	P1(5)
			P2(4)
				P1(3)
				P2(4)
				P3(2)
					P2(4)
					P3(2)
						P3(4)	

9/10

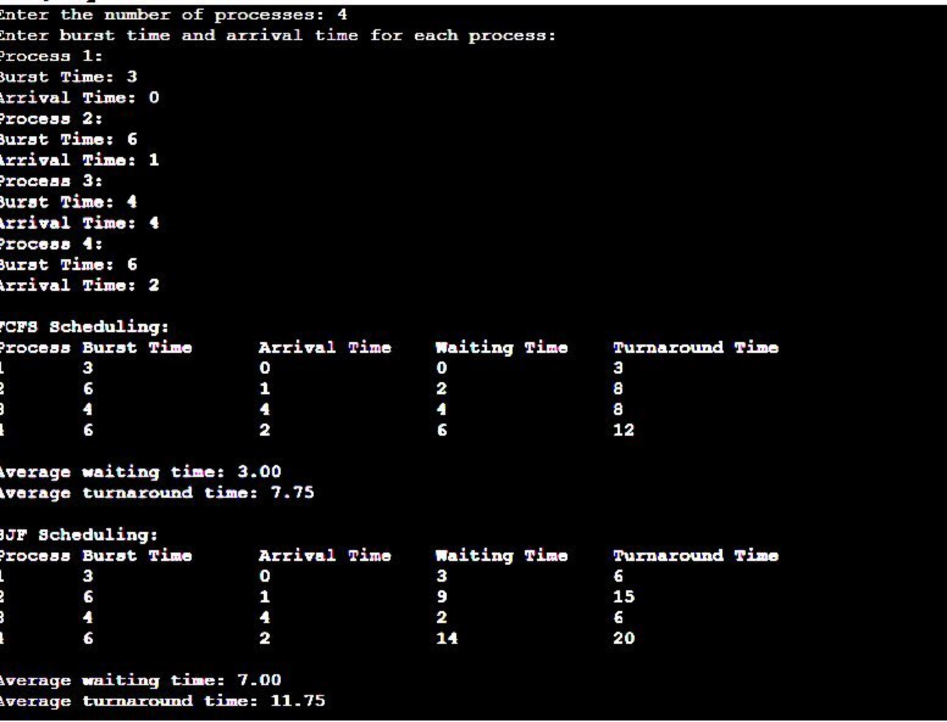
26/6/23

3. SRTF

P0	P0	P1	P2	P2	P3	P1	
0	1	3	4	6	8	10	15

P0(3)	P0(2)
	P1(6)
		P1(6)	P1(5)
			P2(4)
				P2(2)
				P1(5)
				P3(2)
					P1(5)
					P3(2)
						P1(5)	

Show complete tracing



```
Enter the number of processes: 4
Enter burst time and arrival time for each process:
Process 1:
Burst Time: 3
Arrival Time: 0
Process 2:
Burst Time: 6
Arrival Time: 1
Process 3:
Burst Time: 4
Arrival Time: 4
Process 4:
Burst Time: 6
Arrival Time: 2

FCFS Scheduling:
Process Burst Time      Arrival Time    Waiting Time    Turnaround Time
1       3               0               0               3
2       6               1               2               8
3       4               4               4               8
4       6               2               6               12

Average waiting time: 3.00
Average turnaround time: 7.75

SJF Scheduling:
Process Burst Time      Arrival Time    Waiting Time    Turnaround Time
1       3               0               3               6
2       6               1               9               15
3       4               4               2               6
4       6               2               14              20

Average waiting time: 7.00
Average turnaround time: 11.75
```