Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time for Shortest Job First (SJF).

PREEMPTIVE:

```
#include <stdio.h>
#include <limits.h>

void findWaitingTime(int n, int at[], int bt[], int wt[]) {
int rt[n]; // Remaining time array
for (int i = 0; i < n; i++)
rt[i] = bt[i];

int complete = 0, t = 0, min_index, min_rt;
int finish_time;
int shortest = -1;
int check = 0;

while (complete != n) {
min_rt = INT_MAX;
for (int j = 0; j < n; j++) {
if (at[j] <= t && rt[j] > 0 && rt[j] < min_rt) {
min_rt = rt[j];
shortest = j;
check = 1;
}
}

if (check == 0) {
t++;
continue;
}

rt[shortest]--;
min_rt = rt[shortest];

if (rt[shortest] == 0) {
complete++;
finish_time = t + 1;
wt[shortest] = finish_time - at[shortest] - bt[shortest];

if (wt[shortest] < 0)
wt[shortest] = 0;
}
t++;
}
}

void findTurnAroundTime(int n, int at[], int bt[], int wt[], int tat[]) {
for (int i = 0; i < n; i++)
tat[i] = bt[i] + wt[i];
```

```
}
void findAvgTime(int n, int at[], int bt[]) {
int wt[n], tat[n];
float total_wt = 0, total_tat = 0;

findWaitingTime(n, at, bt, wt);
findTurnAroundTime(n, at, bt, wt, tat);

printf("\nProcess\tAT\tBT\tWT\tTAT\n");
for (int i = 0; i < n; i++) {
total_wt += wt[i];
total_tat += tat[i];
printf("%d\t%d\t%d\t%d\t%d\n", i + 1, at[i], bt[i], wt[i], tat[i]);
}

printf("\nAverage Waiting Time = %.2f", total_wt / n);
printf("\nAverage Turnaround Time = %.2f\n", total_tat / n);
}
int main() {
int n;
printf("Enter the number of processes: ");
scanf("%d", &n);

int at[n], bt[n];
printf("Enter Arrival Time and Burst Time for each process:\n");
for (int i = 0; i < n; i++) {
printf("Process %d: ", i + 1);
scanf("%d %d", &at[i], &bt[i]);
}
findAvgTime(n, at, bt);

return 0;
}
```

OUTPUT:

```
Enter the number of processes: 5
Enter Arrival Time and Burst Time for each process:
Process 1: 1 2
Process 2: 3 4
Process 3: 5 6
Process 4: 7 8
Process 5: 9 10

Process AT      BT      WT      TAT
1       1       2       0       2
2       3       4       0       4
3       5       6       2       8
4       7       8       6       14
5       9       10      12      22

Average Waiting Time = 4.00
Average Turnaround Time = 10.00

Process returned 0 (0x0)   execution time : 25.531 s
Press any key to continue.
```

NON-PREEMPTIVE:

```
#include <stdio.h>
#include <limits.h>

void sjfNonPreemptive(int n, int at[], int bt[], int p[]) {
int ct[n], tat[n], wt[n], completed = 0, currentTime = 0, isCompleted[n];

for (int i = 0; i < n; i++)
isCompleted[i] = 0;

while (completed < n) {
int shortest = -1, minBT = INT_MAX;

for (int i = 0; i < n; i++) {
if (at[i] <= currentTime && isCompleted[i] == 0) {
if (bt[i] < minBT || (bt[i] == minBT && (shortest == -1 || at[i] < at[shortest]))) {
minBT = bt[i];
shortest = i;
}
}
}

if (shortest == -1) {
int nextArrival = INT_MAX;
for (int i = 0; i < n; i++) {
if (!isCompleted[i] && at[i] < nextArrival) {
nextArrival = at[i];
}
}
currentTime = nextArrival;
} else {
ct[shortest] = currentTime + bt[shortest];
tat[shortest] = ct[shortest] - at[shortest];
wt[shortest] = tat[shortest] - bt[shortest];
isCompleted[shortest] = 1;
currentTime = ct[shortest];
completed++;
}
}

printf("\nProcess\tAT\tBT\tCT\tTAT\tWT\n");
float totalWT = 0, totalTAT = 0;
for (int i = 0; i < n; i++) {
printf("%d\t%d\t%d\t%d\t%d\t%d\n", p[i], at[i], bt[i], ct[i], tat[i], wt[i]);
totalWT += wt[i];
totalTAT += tat[i];
}

printf("\nAverage Waiting Time: %.2f", totalWT / n);
printf("\nAverage Turnaround Time: %.2f\n", totalTAT / n);
```

```
}

int main() {
int n;
printf("Enter the number of processes: ");
scanf("%d", &n);

int at[n], bt[n], p[n];

printf("Enter Arrival Time and Burst Time for each process:\n");
for (int i = 0; i < n; i++) {
p[i] = i + 1;
printf("Process %d: ", i + 1);
scanf("%d %d", &at[i], &bt[i]);
}

sjfNonPreemptive(n, at, bt, p);

return 0;
}
```

OUTPUT:

```
Enter the number of processes: 5
Enter Arrival Time and Burst Time for each process:
Process 1: 1 2
Process 2: 3 4
Process 3: 5 6
Process 4: 7 8
Process 5: 9 10

Process AT      BT      CT      TAT     WT
1       1       2       3       2       0
2       3       4       7       4       0
3       5       6       13      8       2
4       7       8       21      14      6
5       9       10      31      22      12

Average Waiting Time: 4.00
Average Turnaround Time: 10.00

Process returned 0 (0x0)   execution time : 32.204 s
Press any key to continue.
```