```c
#include <stdio.h>

struct Process {
    int pid;       // Process ID
    int arrival;   // Arrival time
    int burst;     // Burst time
    int remaining;  // Remaining time
    int completion; // Completion time
    int waiting;   // Waiting time
    int turnaround; // Turnaround time
};

int main() {
    int n, completed = 0, current_time = 0, shortest = -1, min_remaining;
    float total_wait = 0, total_turnaround = 0;

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

    struct Process p[n];

    for (int i = 0; i < n; i++) {
        p[i].pid = i + 1;
        printf("\nEnter Arrival Time for P%d: ", i + 1);
        scanf("%d", &p[i].arrival);
        printf("Enter Burst Time for P%d: ", i + 1);
        scanf("%d", &p[i].burst);
        p[i].remaining = p[i].burst;
    }

    printf("\n--- Shortest Job First (Preemptive) Scheduling ---\n");
```

```
while (completed != n) {
    shortest = -1;
    min_remaining = 9999;

    for (int i = 0; i < n; i++) {
        if (p[i].arrival <= current_time && p[i].remaining > 0 && p[i].remaining < min_remaining) {
            min_remaining = p[i].remaining;
            shortest = i;
        }
    }

    if (shortest == -1) {
        current_time++;
        continue;
    }

    p[shortest].remaining--;
    current_time++;

    if (p[shortest].remaining == 0) {
        completed++;
        p[shortest].completion = current_time;
        p[shortest].turnaround = p[shortest].completion - p[shortest].arrival;
        p[shortest].waiting = p[shortest].turnaround - p[shortest].burst;
        total_wait += p[shortest].waiting;
        total_turnaround += p[shortest].turnaround;
    }
}

printf("\nPID\tAT\tBT\tCT\tTAT\tWT\n");
```

```
    for (int i = 0; i < n; i++) {
        printf("P%d\t%d\t%d\t%d\t%d\t%d\n", p[i].pid, p[i].arrival, p[i].burst,
               p[i].completion, p[i].turnaround, p[i].waiting);
    }

    printf("\nAverage Turnaround Time = %.2f", total_turnaround / n);
    printf("\nAverage Waiting Time = %.2f\n", total_wait / n);

    return 0;
}
```

Input

Enter number of processes: 3

Enter Arrival Time for P1: 0

Enter Burst Time for P1: 7

Enter Arrival Time for P2: 2

Enter Burst Time for P2: 4

Enter Arrival Time for P3: 4

Enter Burst Time for P3: 1

output

--- Shortest Job First (Preemptive) Scheduling ---

PID	AT	BT	CT	TAT	WT
P1	0	7	12	12	5
P2	2	4	7	5	1
P3	4	1	5	1	0

Average Turnaround Time = 6.00

Average Waiting Time = 2.00

Part A — SJF Preemptive (Shortest Remaining Time First)

Step 1: Process details

We have:

P1: AT=0, BT=7

P2: AT=2, BT=4

P3: AT=4, BT=1

Step 2: Timeline simulation (Gantt Chart)

Let's go second-by-second ⏳

Time	Ready Processes	Process Chosen	Remarks
0–1	P1	P1	Only P1 available
1–2	P1	P1	Still only P1
2	P1(remaining=5), P2(arrives)	P2 (since 4 < 5)	P2 preempts P1
2–3	P2	Running	
3–4	P2	Running	
4	P2(remaining=2), P3(arrives)	P3 (since 1 < 2)	P3 preempts P2
4–5	P3	Completes at 5	
5–7	P2	Finishes at 7	
7–12	P1	Finishes at 12	

✅ **Completion Times (CT):**

- P1 = 12
- P2 = 7
- P3 = 5

Step 3: Calculate Turnaround Time (TAT) and Waiting Time (WT)

Formula:

- **TAT = CT − AT**
- **WT = TAT − BT**

Process	AT	BT	CT	TAT = CT−AT	WT = TAT−BT
P1	0	7	12	12	5
P2	2	4	7	5	1
P3	4	1	5	1	0

Step 4: Average Times

- **Average TAT = (12 + 5 + 1) / 3 = 6.00**
- **Average WT = (5 + 1 + 0) / 3 = 2.00**

✅ **Final Paper Answer (SJF Preemptive)**

PID	AT	BT	CT	TAT	WT
P1	0	7	12	12	5
P2	2	4	7	5	1
P3	4	1	5	1	0

Average TAT = 6.00
Average WT = 2.00