```c
#include <stdio.h>

#define MAX 100

struct Process {
    int id, at, bt, rt, wt, tat, ct;
};

void sjf_preemptive(struct Process p[], int n) {
    int completed = 0, time = 0, shortest = -1;
    int min_bt = 9999;

    printf("\n--- SJF Preemptive Scheduling ---\n");

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

        for (int i = 0; i < n; i++) {
            if (p[i].at <= time && p[i].rt > 0 && p[i].rt < min_bt) {
                min_bt = p[i].rt;
                shortest = i;
            }
        }

        if (shortest == -1) {
            time++;
            continue;
```

```
        }

        p[shortest].rt--;
        time++;

        if (p[shortest].rt == 0) {
            completed++;
            p[shortest].ct = time;
            p[shortest].tat = p[shortest].ct - p[shortest].at;
            p[shortest].wt = p[shortest].tat - p[shortest].bt;
        }
    }

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

void round_robin(struct Process p[], int n, int tq) {
    int time = 0, completed = 0;
    int done;

    printf("\n--- Round Robin Scheduling (TQ = %d) ---\n", tq);

    while (completed != n) {
        done = 1;
```

```
        for (int i = 0; i < n; i++) {
            if (p[i].rt > 0 && p[i].at <= time) {
                done = 0;
                if (p[i].rt > tq) {
                    time += tq;
                    p[i].rt -= tq;
                } else {
                    time += p[i].rt;
                    p[i].ct = time;
                    p[i].tat = p[i].ct - p[i].at;
                    p[i].wt = p[i].tat - p[i].bt;
                    p[i].rt = 0;
                    completed++;
                }
            } else if (p[i].rt > 0 && p[i].at > time) {
                time++;
            }
        }

        if (done)
            break;
    }

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

```
int main() {
    struct Process p[MAX];
    int n, choice, tq;

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

    for (int i = 0; i < n; i++) {
        p[i].id = i + 1;
        printf("Enter AT and BT for process %d: ", i + 1);
        scanf("%d%d", &p[i].at, &p[i].bt);
        p[i].rt = p[i].bt;  // Initialize remaining time
    }

    printf("\nChoose Scheduling Algorithm:\n1. SJF (Preemptive)\n2. Round Robin\nEnter choice:
");
    scanf("%d", &choice);

    if (choice == 1) {
        sjf_preemptive(p, n);
    } else if (choice == 2) {
        printf("Enter Time Quantum: ");
        scanf("%d", &tq);
        round_robin(p, n, tq);
    } else {
        printf("Invalid choice.\n");
    }
```

```
    return 0;
}
```