

```
#include <stdio.h>

struct process {
    int pid, at, bt, pr;
    int ct, tat, wt, done;
};

int main() {
    int n;
    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("Enter AT, BT, PRIORITY for P%d: ", i + 1);
        scanf("%d %d %d", &p[i].at, &p[i].bt, &p[i].pr);
        p[i].done = 0;
    }

    int completed = 0, current_time = 0;
    float totalTAT = 0, totalWT = 0;

    while (completed < n) {
        int idx = -1;
        int best_pr = 9999;

        // Find process with highest priority (smallest pr value)
        for (int i = 0; i < n; i++) {
            if (!p[i].done && p[i].at <= current_time) {
                if (p[i].pr < best_pr) {
                    best_pr = p[i].pr;
                    idx = i;
                }
            }
        }

        // If no process has arrived, move time forward
        if (idx == -1) {
            current_time++;
            continue;
        }

        // Execute chosen process
        current_time += p[idx].bt;
        p[idx].ct = current_time;
        p[idx].tat = p[idx].ct - p[idx].at;
        p[idx].wt = p[idx].tat - p[idx].bt;

        totalTAT += p[idx].tat;
        totalWT += p[idx].wt;

        p[idx].done = 1;
        completed++;
    }

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

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

```
    return 0;  
}
```