

```

#include <stdio.h>

typedef struct process {
    int pid;
    int prior;
    int AT;
    int BT;
    int CT;
    int TAT;
    int WT;
    int mark;
    int RT;
    int b;
} PS;

PS P[100], temp;
int n;

int main() {
    printf("Enter the number of processors:\n");
    scanf("%d", &n);
    printf("Enter PID, Priority, Arrival time, and burst time:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d%d%d%d", &P[i].pid, &P[i].prior, &P[i].AT, &P[i].BT);
        P[i].mark = 0;
        P[i].RT = -1;
        P[i].b=P[i].BT;
    }
    int min = P[0].prior;
    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            if (P[i].prior > P[j].prior) {
                temp = P[i];
                P[i] = P[j];
                P[j] = temp;
            }
        }
    }
    double ttat = 0.0, twt = 0.0;
    int count = 0, pro = 0, priority = P[0].prior;
    int x = 0, c = 0;
    while (count < n) {
        int minPriority = 9999;
        for (int i = 0; i < n; i++) {
            if (P[i].AT <= c && P[i].prior <= minPriority && P[i].b> 0) {
                x = i;
                minPriority = P[i].prior;
            }
        }
        if (P[x].RT == -1) {

```

```

        P[x].RT = c - P[x].AT;
    }
    P[x].b--;
    c++;
    if (P[x].b == 0) {
        count++;
        P[x].CT = c;
        P[x].mark = 1;
    }
}
for (int i = 0; i < n; i++) {
    P[i].TAT = P[i].CT - P[i].AT;
    P[i].WT = P[i].TAT - P[i].BT;
}
printf("Priority Scheduling (pre-emptive)\n");
printf("PID\tPrior\tAT\tBT\tCT\tTAT\tWT\tRT\n");
for (int i = 0; i < n; i++) {
    printf("%d\t%d\t\t\t%d\t%d\t%d\t%d\t%d\t%d\n", P[i].pid, P[i].prior, P[i].AT,
P[i].BT, P[i].CT, P[i].TAT, P[i].WT, P[i].RT);
    ttat += P[i].TAT;
    twt += P[i].WT;
}
printf("\nAverage turnaround time is %lf ms\n", ttat / (double)n);
printf("Average waiting time is %lf ms\n", twt / (double)n);
return 0;
}

```