

## 5.PRIORITY SCHEDULING

```
#include <stdio.h>

struct Process {
    int id;
    int burst;
    int priority;
    int waiting;
    int turnaround;
};

int main() {
    int n;
    printf("Enter number of processes: ");
    scanf("%d", &n);
    struct Process p[n];
    float avg_wt = 0, avg_tat = 0;
    for (int i = 0; i < n; i++) {
        p[i].id = i + 1;
        printf("Enter burst time for P%d: ", i + 1);
        scanf("%d", &p[i].burst);
        printf("Enter priority for P%d: ", i + 1);
        scanf("%d", &p[i].priority);
    }
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (p[j].priority < p[i].priority) {
                struct Process temp = p[i];
                p[i] = p[j];
                p[j] = temp;
            }
        }
    }
}
```

```

    }
}
p[0].waiting = 0;
for (int i = 1; i < n; i++) {
    p[i].waiting = p[i - 1].waiting + p[i - 1].burst;
}
for (int i = 0; i < n; i++) {
    p[i].turnaround = p[i].waiting + p[i].burst;
    avg_wt += p[i].waiting;
    avg_tat += p[i].turnaround;
}
avg_wt /= n;
avg_tat /= n;
printf("\nProcess\tPriority\tBurst\tWaiting\tTurnaround\n");
for (int i = 0; i < n; i++) {
    printf("P%d\t%d\t%d\t%d\t%d\n",
        p[i].id, p[i].priority, p[i].burst, p[i].waiting, p[i].turnaround);
}

printf("\nAverage Waiting Time: %.2f", avg_wt);
printf("\nAverage Turnaround Time: %.2f\n", avg_tat);

return 0;
}

```