

Program: Priority Scheduling With Arrival Time.

CODE:

```
#include <iostream> using namespace std;

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

// Comparator function to sort by priority (lower value = higher
// priority) bool compare(Process a, Process b) { if (a.at == b.at)
return a.priority < b.priority; return a.at < b.at; }

void priorityScheduling(Process p[], int n) { sort(p, p + n, compare);
// Sort based on arrival time and priority

int time = 0;
float total_wt = 0, total_tat = 0;

for (int i = 0; i < n; i++) {
    if (time < p[i].at) {
        time = p[i].at; // Ensure CPU is idle until process arrives
    }
    p[i].wt = time - p[i].at;
    p[i].tat = p[i].wt + p[i].bt;
    p[i].ct = time + p[i].bt;

    time += p[i].bt;

    total_wt += p[i].wt;
    total_tat += p[i].tat;
}

cout << "\nProcess\tAT\tBT\tPriority\tWT\tTAT\tCT\n";
for (int i = 0; i < n; i++) {
    cout << p[i].id << "\t" << p[i].at << "\t" << p[i].bt << "\t" <<
p[i].priority
    << "\t\t" << p[i].wt << "\t" << p[i].tat << "\t" << p[i].ct
    << endl;
}

cout << "\nAverage Waiting Time: " << (total_wt / n);
```

```

cout << "\nAverage Turnaround Time: " << (total_tat / n) << endl;
}

int main() { int n; cout << "Enter number of processes: "; cin >> n;

Process p[n];

for (int i = 0; i < n; i++) {
    cout << "Enter arrival time, burst time, and priority for process
" << i + 1 << ": ";
    cin >> p[i].at >> p[i].bt >> p[i].priority;
    p[i].id = i + 1;
}

priorityScheduling(p, n);
return 0;
}

```

OUTPUT:

Enter number of processes: 3

Enter arrival time, burst time, and priority for process 1: 0 5 2

Enter arrival time, burst time, and priority for process 2: 1 3 1

Enter arrival time, burst time, and priority for process 3: 2 8 3

Process	AT	BT	Priority	WT	TAT	CT
1	0	5	2	0	5	5
2	1	3	1	4	7	8
3	2	8	3	6	14	16

Average Waiting Time: 3.33333

Average Turnaround Time: 8.66667

Conclusion: The program Priority scheduling was implemented successfully.