26. Write a program to compute the average waiting time and average turnaround time based on Priority scheduling for the following process with the given CPU burst times (and the assumption that all jobs arrive at the same time.)

Process Burst Time Priority

P1 30 2

P2 5 1

P3 12 3

Program:-

#include <stdio.h>

#define MAX\_PROCESSES 3

// Structure to represent a process

typedef struct {

int burstTime;

int priority;

int waitingTime;

int turnaroundTime;

} Process;

// Function to calculate average waiting time and average turnaround time

void calculateAverageTimes(Process processes[], int numProcesses, float \*avgWaitingTime, float \*avgTurnaroundTime) {

int totalWaitingTime = 0;

int totalTurnaroundTime = 0;

// Calculate waiting time for each process

processes[0].waitingTime = 0;

for (int i = 1; i < numProcesses; i++) {

processes[i].waitingTime = processes[i - 1].burstTime + processes[i - 1].waitingTime;

totalWaitingTime += processes[i].waitingTime;

}

// Calculate turnaround time for each process

for (int i = 0; i < numProcesses; i++) {

processes[i].turnaroundTime = processes[i].burstTime + processes[i].waitingTime;

totalTurnaroundTime += processes[i].turnaroundTime;

}

// Calculate average waiting time and average turnaround time

\*avgWaitingTime = (float)totalWaitingTime / numProcesses;

\*avgTurnaroundTime = (float)totalTurnaroundTime / numProcesses;

}

int main() {

Process processes[MAX\_PROCESSES] = {

{30, 2, 0, 0}, // P1

{5, 1, 0, 0}, // P2

{12, 3, 0, 0} // P3

};

int numProcesses = sizeof(processes) / sizeof(processes[0]);

float avgWaitingTime, avgTurnaroundTime;

calculateAverageTimes(processes, numProcesses, &avgWaitingTime, &avgTurnaroundTime);

printf("Process\tBurst Time\tPriority\tWaiting Time\tTurnaround Time\n");

for (int i = 0; i < numProcesses; i++) {

printf("P%d\t\t%d\t\t%d\t\t%d\t\t%d\n", i + 1, processes[i].burstTime, processes[i].priority,

processes[i].waitingTime, processes[i].turnaroundTime);

}

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

printf("Average Turnaround Time: %.2f\n", avgTurnaroundTime);

return 0;

}

Output:-

