Practical 10

# Aim : Write a program to calculate total waiting time and turn around time of n processes with a priority scheduling algorithm .

**Code:**

#include<stdio.h> struct priority\_scheduling { char process\_name; int burst\_time; int waiting\_time; int turn\_around\_time;

int priority;

};

int main() { int number\_of\_process; int total = 0; struct priority\_scheduling temp\_process; int ASCII\_number = 65;

int position; float average\_waiting\_time; float average\_turnaround\_time; printf("Enter the total number of Processes: "); scanf("%d", & number\_of\_process); struct priority\_scheduling process[number\_of\_process]; printf("\nPlease Enter the Burst Time and Priority of each process:\n"); for (int i = 0; i < number\_of\_process; i++)

{ process[i].process\_name = (char) ASCII\_number; printf("\nEnter the details of the process %c \n", process[i].process\_name); printf("Enter the burst time: "); scanf("%d", & process[i].burst\_time); printf("Enter the priority: "); scanf("%d", & process[i].priority); ASCII\_number++;

}

for (int i = 0; i < number\_of\_process; i++)

{ position = i; for (int j = i + 1; j < number\_of\_process; j++) {

if (process[j].priority > process[position].priority) position = j;

}

temp\_process = process[i]; process[i] = process[position]; process[position] = temp\_process;

}

process[0].waiting\_time = 0; for (int i = 1; i

< number\_of\_process; i++)

{ process[i].waiting\_time = 0; for (int j = 0; j

< i; j++) { process[i].waiting\_time += process[j].burst\_time;

}

total += process[i].waiting\_time;

}

average\_waiting\_time = (float) total / (float) number\_of\_process; total = 0; printf("\n\nProcess\_name \t Burst Time \t Waiting Time \t Turnaround Time\n");

printf(" \n");

for (int i = 0; i < number\_of\_process; i++) { process[i].turn\_around\_time = process[i].burst\_time + process[i].waiting\_time; total += process[i].turn\_around\_time;

printf("\t %c \t\t %d \t\t %d \t\t %d", process[i].process\_name, process[i].burst\_time, process[i].waiting\_time, process[i].turn\_around\_time);

printf("\n \n");

}

average\_turnaround\_time = (float) total / (float) number\_of\_process; printf("\n\n Average Waiting Time : %f", average\_waiting\_time); printf("\n Average Turnaround Time: %f\n", average\_turnaround\_time); return 0;

}

# Output:

