3.Write a program to compute the average waiting time and average turnaround time based on Non Preemptive Shortest-Job-First 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

P1 6

P2 8

P3 7

P4 3

PROGRAM :

#include <stdio.h>

#define MAX\_PROCESSES 10

int main() {

int n;

int burst\_times[MAX\_PROCESSES];

int waiting\_times[MAX\_PROCESSES];

int turnaround\_times[MAX\_PROCESSES];

int i, j, temp;

\ // read the number of processes and their burst times

printf("Enter the number of processes: ");

scanf("%d", &n);

printf("Enter the CPU burst times:\n");

for (i = 0; i < n; i++) {

scanf("%d", &burst\_times[i]);

}

for (i = 0; i < n - 1; i++)

int min\_idx = i;

for (j = i + 1; j < n; j++) {

if (burst\_times[j] < burst\_times[min\_idx]) {

min\_idx = j;

}

}

temp = burst\_times[i];

burst\_times[i] = burst\_times[min\_idx];

burst\_times[min\_idx] = temp;

}

waiting\_times[0] = 0;

turnaround\_times[0] = burst\_times[0];

for (i = 1; i < n; i++) {

waiting\_times[i] = waiting\_times[i-1] + burst\_times[i-1];

turnaround\_times[i] = turnaround\_times[i-1] + burst\_times[i];

}

float avg\_waiting\_time = 0.0;

float avg\_turnaround\_time = 0.0;

for (i = 0; i < n; i++) {

avg\_waiting\_time += waiting\_times[i];

avg\_turnaround\_time += turnaround\_times[i];

}

avg\_waiting\_time /= n;

avg\_turnaround\_time /= n;

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

for (i = 0; i < n; i++) {

printf("%d\t%d\t\t%d\t\t%d\n", i+1, burst\_times[i], waiting\_times[i], turnaround\_times[i]);

}

printf("Average waiting time: %.2f\n", avg\_waiting\_time);

printf("Average turnaround time: %.2f\n", avg\_turnaround\_time);

return 0;

}

OUTPUT :

