**EXPERIMENT-4**

4. Construct a scheduling program with C that selects the waiting process with the smallest

execution time to execute next.

#include <stdio.h>

int main() {

int n, i, j;

int burstTime[100], process[100], waitingTime[100], turnaroundTime[100];

int totalWaitingTime = 0, totalTurnaroundTime = 0;

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

scanf("%d", &n);

printf("Enter burst time for each process:\n");

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

printf("Process %d: ", i + 1);

scanf("%d", &burstTime[i]);

process[i] = i + 1;

}

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

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

if (burstTime[i] > burstTime[j]) {

// Swap burst times

int temp = burstTime[i];

burstTime[i] = burstTime[j];

burstTime[j] = temp;

int t = process[i];

process[i] = process[j];

process[j] = t;

}

}

}

waitingTime[0] = 0;

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

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

}

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

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

totalWaitingTime += waitingTime[i];

totalTurnaroundTime += turnaroundTime[i];

}

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

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

printf("P%d\t%d\t\t%d\t\t%d\n", process[i], burstTime[i], waitingTime[i], turnaroundTime[i]);

}

float avgWaitingTime = (float)totalWaitingTime / n;

float avgTurnaroundTime = (float)totalTurnaroundTime / n;

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

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

return 0;

}

**SAMPLE INPUT**

Enter the number of processes: 4

Enter burst time for each process:

Process 1: 6

Process 2: 8

Process 3: 7

Process 4: 3

SAMPLE OUTPUT

Process Burst Time Waiting Time Turnaround Time

P4 3 0 3

P1 6 3 9

P3 7 9 16

P2 8 16 24

Average Waiting Time: 7.00

Average Turnaround Time: 13.00