Write a C program to simulate the following CPU scheduling algorithm to find turnaround time and waiting time.

b) SJF(Pre-emptive and Non - pre-emptive)

**Pre-emptive :**

#include <stdio.h>

int arr[5], bt[5], remaining\_bt[5], wt[5], tat[5], pid[5], ct[5];

int totalwt = 0, totaltat = 0, time = 0, completed = 0;

int n = 5;

void main() {

printf("Enter the Arrival times for 5 processes:\n");

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

pid[i] = i + 1;

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

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

}

printf("Enter the Burst times for 5 processes:\n");

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

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

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

remaining\_bt[i] = bt[i];

}

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

wt[i] = 0;

tat[i] = 0;

}

while (completed != n) {

int idx = -1;

int min\_bt = 9999;

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

if (arr[i] <= time && remaining\_bt[i] > 0) {

if (remaining\_bt[i] < min\_bt) {

min\_bt = remaining\_bt[i];

idx = i;

} else if (remaining\_bt[i] == min\_bt) {

// If burst time is the same, choose the one with the earlier arrival time

if (arr[i] < arr[idx]) {

idx = i;

}

}

}

}

if (idx != -1) {

remaining\_bt[idx]--;

time++;

if (remaining\_bt[idx] == 0) {

completed++;

ct[idx] = time;

tat[idx] = ct[idx] - arr[idx];

wt[idx] = tat[idx] - bt[idx];

totalwt += wt[idx];

totaltat += tat[idx];

}

} else {

time++;

}

}

printf("\nProcess ID | Arrival Time | Burst Time | Waiting Time | Turnaround Time | Completion Time\n");

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

printf(" %d | %d | %d | %d | %d | %d\n",

pid[i], arr[i], bt[i], wt[i], tat[i], ct[i]);

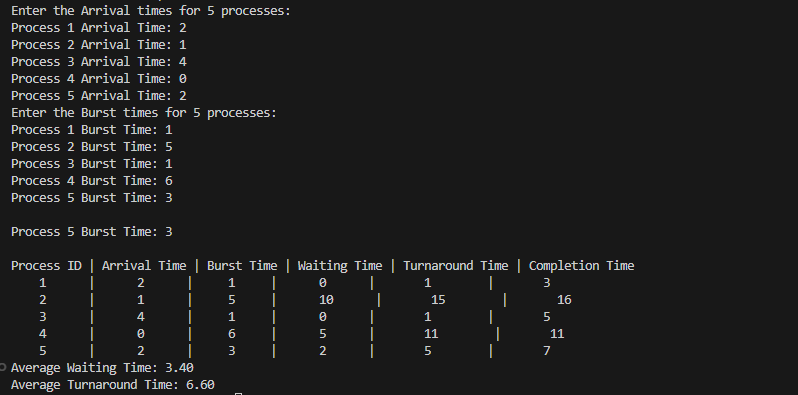
}

printf("Average Waiting Time: %.2f\n", (float)totalwt / n);

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

}

Output :



**Non Pre-emptive :**

#include <stdio.h>

int arr[5], bt[5], wt[5], tat[5], pid[5], ct[5];

int totalwt = 0, totaltat = 0, n = 5;

void main() {

printf("Enter the Arrival times for 5 processes:\n");

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

pid[i] = i + 1;

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

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

}

printf("Enter the Burst times for 5 processes:\n");

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

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

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

}

int completed = 0, time = 0;

int is\_completed[5] = {0};

while (completed != n) {

int idx = -1, min\_bt = 9999;

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

if (arr[i] <= time && !is\_completed[i] && bt[i] < min\_bt) {

min\_bt = bt[i];

idx = i;

}

}

if (idx != -1) {

time += bt[idx];

ct[idx] = time;

tat[idx] = ct[idx] - arr[idx];

wt[idx] = tat[idx] - bt[idx];

totalwt += wt[idx];

totaltat += tat[idx];

is\_completed[idx] = 1;

completed++;

} else {

time++;

}

}

printf("\nProcess ID | Arrival Time | Burst Time | Waiting Time | Turnaround Time | Completion Time\n");

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

printf(" %d | %d | %d | %d | %d | %d\n",

pid[i], arr[i], bt[i], wt[i], tat[i], ct[i]);

}

printf("Average Waiting Time: %.2f\n", (float)totalwt / n);

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

}

Output :

