**LAB ASSIGNMENT- 5**

**Aim:** To create C programs for the different scheduling algorithms.

**To perform**: Create and execute C programs for following CPU Scheduling Algorithms:

* First Come First Serve (FCFS)
* Shortest Job First (SJF)
* Round Robin Scheduling

**To submit:**

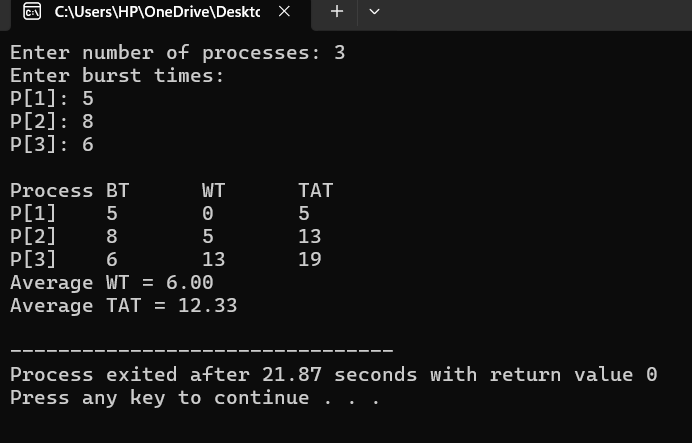
**C Codes for the above scheduling algorithms with their outputs:**

**1. First Come First Serve (FCFS)**

**Code:**

#include <stdio.h>  
  
int main() {  
 int n, i;  
 printf("Enter number of processes: ");  
 scanf("%d", &n);  
 int bt[n], wt[n], tat[n];  
 float avg\_wt = 0, avg\_tat = 0;  
  
 printf("Enter burst times:\n");  
 for(i = 0; i < n; i++) {  
 printf("P[%d]: ", i+1);  
 scanf("%d", &bt[i]);  
 }  
  
 wt[0] = 0;  
 for(i = 1; i < n; i++) {  
 wt[i] = bt[i-1] + wt[i-1];  
 }  
  
 for(i = 0; i < n; i++) {  
 tat[i] = bt[i] + wt[i];  
 avg\_wt += wt[i];  
 avg\_tat += tat[i];  
 }  
  
 printf("\nProcess\tBT\tWT\tTAT\n");  
 for(i = 0; i < n; i++) {  
 printf("P[%d]\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);  
 }  
  
 printf("Average WT = %.2f\n", avg\_wt/n);  
 printf("Average TAT = %.2f\n", avg\_tat/n);  
  
 return 0;  
}

**Sample Output:**

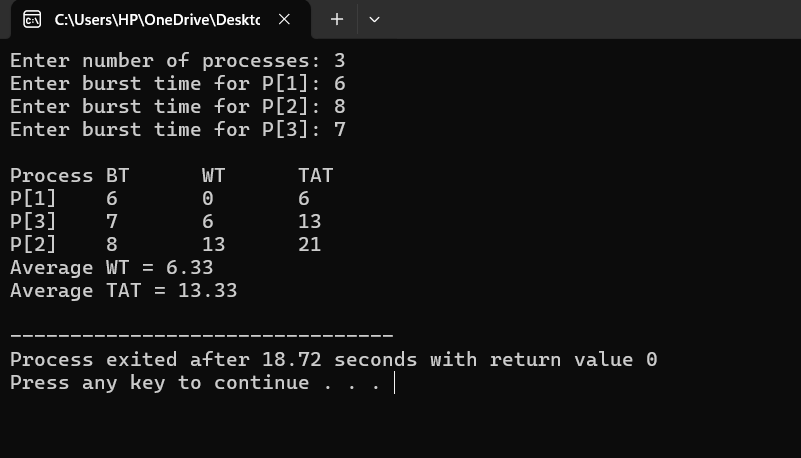


**2. Shortest Job First (SJF)**

**Code:**

#include <stdio.h>  
  
int main() {  
 int n, i, j;  
 printf("Enter number of processes: ");  
 scanf("%d", &n);  
 int p[n], bt[n], wt[n], tat[n], temp;  
 float avg\_wt = 0, avg\_tat = 0;  
  
 for(i = 0; i < n; i++) {  
 p[i] = i+1;  
 printf("Enter burst time for P[%d]: ", i+1);  
 scanf("%d", &bt[i]);  
 }  
  
 for(i = 0; i < n-1; i++) {  
 for(j = i+1; j < n; j++) {  
 if(bt[i] > bt[j]) {  
 temp = bt[i]; bt[i] = bt[j]; bt[j] = temp;  
 temp = p[i]; p[i] = p[j]; p[j] = temp;  
 }  
 }  
 }  
  
 wt[0] = 0;  
 for(i = 1; i < n; i++) {  
 wt[i] = wt[i-1] + bt[i-1];  
 }  
  
 for(i = 0; i < n; i++) {  
 tat[i] = bt[i] + wt[i];  
 avg\_wt += wt[i];  
 avg\_tat += tat[i];  
 }  
  
 printf("\nProcess\tBT\tWT\tTAT\n");  
 for(i = 0; i < n; i++) {  
 printf("P[%d]\t%d\t%d\t%d\n", p[i], bt[i], wt[i], tat[i]);  
 }  
  
 printf("Average WT = %.2f\n", avg\_wt/n);  
 printf("Average TAT = %.2f\n", avg\_tat/n);  
  
 return 0;  
}

**Sample Output:**



**3. Round Robin (RR)**

**Code:**

#include <stdio.h>  
  
int main() {  
 int i, j, n, time, remain, temps = 0, tq;  
 int wt = 0, tat = 0, at[10], bt[10], rt[10];  
  
 printf("Enter number of processes: ");  
 scanf("%d", &n);  
 remain = n;  
  
 for(i = 0; i < n; i++) {  
 printf("Enter arrival time and burst time for P[%d]: ", i+1);  
 scanf("%d%d", &at[i], &bt[i]);  
 rt[i] = bt[i];  
 }  
  
 printf("Enter time quantum: ");  
 scanf("%d", &tq);  
  
 printf("\nProcess\tTurnaround Time\tWaiting Time\n");  
 for(time = 0, i = 0; remain != 0;) {  
 if(rt[i] > 0 && at[i] <= time) {  
 if(rt[i] <= tq) {  
 time += rt[i];  
 rt[i] = 0;  
 remain--;  
 printf("P[%d]\t\t%d\t\t%d\n", i+1, time - at[i], time - at[i] - bt[i]);  
 wt += time - at[i] - bt[i];  
 tat += time - at[i];  
 } else {  
 rt[i] -= tq;  
 time += tq;  
 }  
 }  
 i = (i + 1) % n;  
 }  
  
 printf("Average Waiting Time = %.2f\n", (float)wt / n);  
 printf("Average Turnaround Time = %.2f\n", (float)tat / n);  
 return 0;  
}

**Sample Output:**

