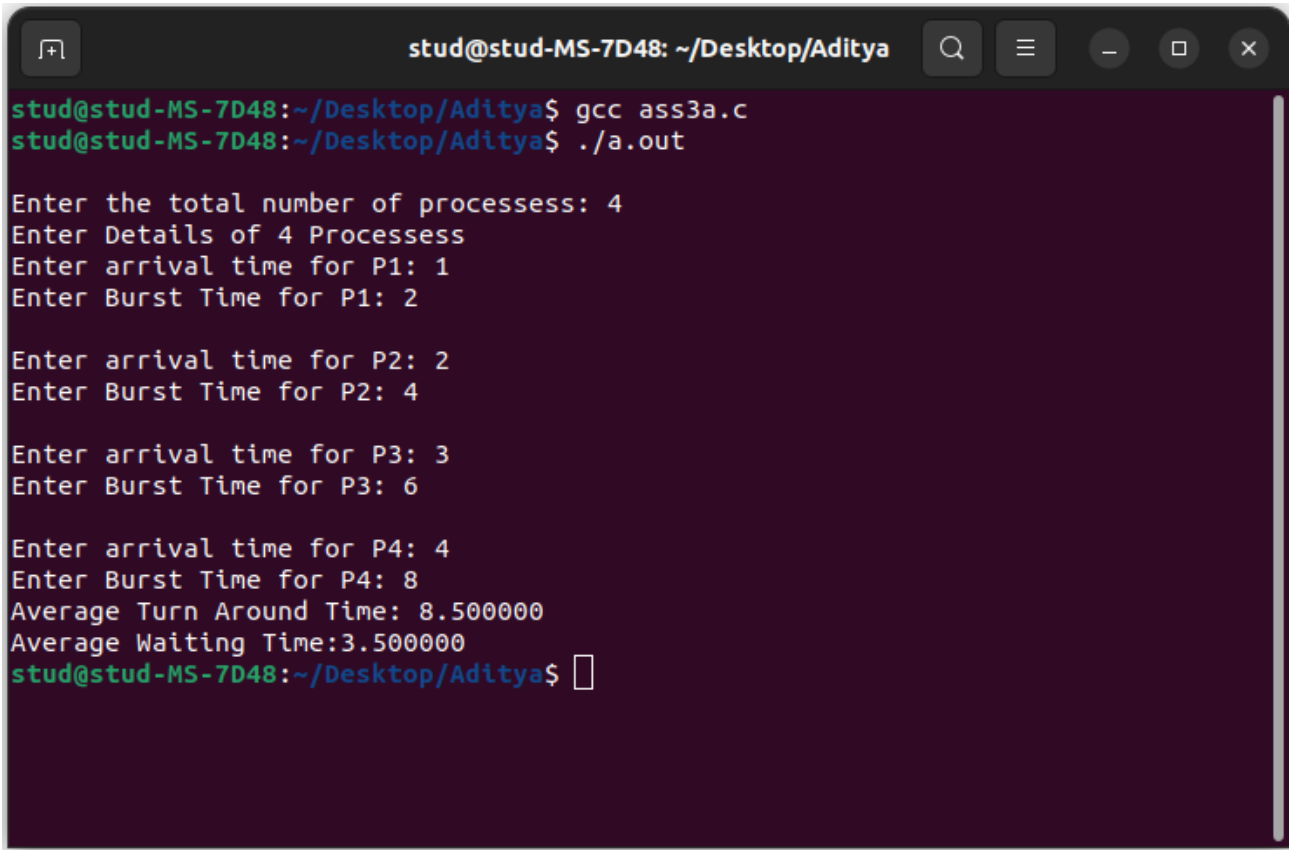
Program :

Shortest Job First:

```
#include<stdio.h>
int arrival_t[100],burst_t[100],ct[100],temp[100];
int main()
{
int i , smallest , count = 0, j,n;
double avg_wt = 0,avg_tat = 0, end;
printf("\nEnter the total number of processess: ");
scanf("%d",&n);
printf("Enter Details of %d Processess",n);
for(i = 0;i<n;i++)
{
printf("\nEnter arrival time for P%d: " , i+1);
scanf("%d",&arrival_t[i]);
printf("Enter Burst Time for P%d: " , i+1);
scanf("%d",&burst_t[i]);
temp[i]=burst_t[i];
}
burst_t[99] = 1000;
for(i =0;count != n; i++)
{
smallest = 99;
for(j = 0;j<n;j++)
{
if (arrival_t[j]<=i && burst_t[j]<burst_t[smallest] &&burst_t[j]>0)
{
smallest =j;
}
}
burst_t[smallest]--;
if(burst_t[smallest] ==0)
{
count++;
}
ct[smallest]=i+1;
}
for(i=0;i<n;i++)
{
int TAT =ct[i]-arrival_t[i];
avg_tat += TAT;
avg_wt +=TAT-temp[i];
}
printf("Average Turn Around Time: %lf\n",avg_tat/n);
printf("Average Waiting Time:%lf\n",avg_wt/n);
return 0;
}
```

Output :



Round Robin :

```
#include <stdio.h>
int arrival_t[100], burst_t[100], remaining_t[100], ct[100];
int main() {
int n, time_quantum;
printf("Enter the total number of processes: ");
scanf("%d", &n);
printf("Enter details of %d processes:\n", n);
for (int i = 0; i < n; i++) {
printf("Enter arrival time for P%d: ", i + 1);
scanf("%d", &arrival_t[i]);
printf("Enter burst time for P%d: ", i + 1);
scanf("%d", &burst_t[i]);
remaining_t[i] = burst_t[i];
}
printf("Enter time quantum: ");
scanf("%d", &time_quantum);
int current_time = 0, count = 0;
while (count < n) {
int done = 0;
for (int i = 0; i < n; i++) {
if (remaining_t[i] > 0 && arrival_t[i] <= current_time) {
done = 1;if (remaining_t[i] > time_quantum) {
current_time += time_quantum;
```

```
remaining_t[i] -= time_quantum;
} else {
current_time += remaining_t[i];
ct[i] = current_time;
remaining_t[i] = 0;
count++;
}
}
}
if (!done) {
current_time++;
}
}
double avg_wt = 0, avg_tat = 0;
for (int i = 0; i < n; i++) {
int TAT = ct[i] - arrival_t[i];
avg_tat += TAT;
avg_wt += TAT - burst_t[i];
}
printf("Average Turnaround Time: %.2f\n", avg_tat / n);
printf("Average Waiting Time: %.2f\n", avg_wt / n);
}
return 0;
```

Output :