# Assignment 6

**Topic:** Simulation of CPU Scheduling

| Sl. No. | Question |
|---|---|
| 1. | Write a menu driven program to implement the following CPU scheduling algorithms.<br>a. First come First Serve<br>b. Round Robin<br>c. Shortest Job First<br>d. Priority Scheduling<br>Find out the average waiting time and average turnaround time in each algorithm. |

**Program for FCFS**

```
#include<stdio.h>
#include<conio.h>
void main()
{
   float process[500],aTime[500],bTime[500],abTime[500],wTime[500],tat_time[500];
   int n = 0,i = 0 ;
   float aw_time = 0, atat_time = 0;
   printf("*** FCFS Scheduling Algorithm Using Arrival Time ***\n");
   printf("\nEnter the number of process : ");
   scanf("%d",&n);

   printf("Enter the Arrival time and Burst time.\n\n");
   printf("\tA_Time B_Time\n");
   for(i = 0 ; i < n ; i++){
      process[i]=i+1;
      printf("P%d :\t", i+1);
      scanf("%f\t%f",&aTime[i],&bTime[i]);
   }
   printf("\n\nProcess\tA_Time\tB_Time\n");
   for(i = 0 ; i < n ; i++){
      printf("P[%d]\t%.2f\t%.2f\n",i,aTime[i],bTime[i]);
   }
   wTime[0] = 0;
   tat_time[0] = bTime[0];
   abTime[0] = bTime[0]+aTime[0];
   for( i = 1 ; i < n ; i++){
      abTime[i] = abTime[i-1] + bTime[i];
      tat_time[i] = abTime[i] - aTime[i];
      wTime[i] = tat_time[i] - bTime[i];
   }
   for(i = 0 ; i < n ; i++){
      aw_time = aw_time + wTime[i];
      atat_time = atat_time + tat_time[i];
   }
   printf("\tA_time\tB_time\tC_time\tTat_time  W_time\n");
   for(i = 0 ; i < n ; i++){
```

```
printf("P[%d]\t%.2f\t%0.2f\t%0.2f\t%0.2f\t%0.2f\n",i,aTime[i],bTime[i],abTime[i],tat_time[i],wTime[i
]);
    }
    printf("\nAverage waiting time : %0.2f",aw_time/n);
    printf("\nAverage turn around time : %0.2f",atat_time/n);
}
```

## Program for Round Robin

```
#include<stdio.h>

int main()
{

  int count,j,n,time,remain,flag=0,time_quantum;
  int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
  printf("Enter Total Process:\t ");
  scanf("%d",&n);
  remain=n;
  for(count=0;count<n;count++)
  {
    printf("Enter Arrival Time and Burst Time for Process Process Number %d :",count+1);
    scanf("%d",&at[count]);
    scanf("%d",&bt[count]);
    rt[count]=bt[count];
  }
  printf("Enter Time Quantum:\t");
  scanf("%d",&time_quantum);
  printf("\n\nProcess\t|Turnaround Time|Waiting Time\n\n");
  for(time=0,count=0;remain!=0;)
  {
    if(rt[count]<=time_quantum && rt[count]>0)
    {
      time+=rt[count];
      rt[count]=0;
      flag=1;
    }
    else if(rt[count]>0)
    {
      rt[count]-=time_quantum;
      time+=time_quantum;
    }
    if(rt[count]==0 && flag==1)
    {
      remain--;
      printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-at[count]-bt[count]);
      wait_time+=time-at[count]-bt[count];
      turnaround_time+=time-at[count];
      flag=0;
    }
    if(count==n-1)
      count=0;
    else if(at[count+1]<=time)
      count++;
    else
      count=0;
  }
  printf("\nAverage Waiting Time= %f\n",wait_time*1.0/n);
```