Process Scheduling:

```
//FIRST COME FIRST SERVE SCHEDULING ALGORITHM
#include <stdio.h>
struct process{
      int pid;
      int bt;
      int wt,tt;
}p[10];

int main(){
      int i,n,totwt,tottt;
      float avgwt,avgtat;

      printf("Enter the no of process \n");
      scanf("%d",&n);
      for(i=1;i<=n;i++){
            p[i].pid=i;
            printf("Enter the burst time for process %d\n",i);
            scanf("%d",&p[i].bt);
      }
      p[1].wt=0;
      p[1].tt=p[1].bt+p[1].wt;
      i=2;
      while(i<=n){
            p[i].wt=p[i-1].bt+p[i-1].wt;
            p[i].tt=p[i].bt+p[i].wt;
            i++;
      }
      i=1;
      totwt=tottt=0;
      printf("\n Processid \t Burst Time\t Waiting Time \t Turn Around
Time\n");
      while(i<=n){
            printf("\n\t%d\t %d \t\t %d \t\t
%d",p[i].pid,p[i].bt,p[i].wt,p[i].tt);
            totwt=p[i].wt+totwt;
            tottt=p[i].tt+tottt;
            i++;
      }
      avgwt=(float)totwt/n;
      avgtat=(float)tottt/n;
      printf("\nAverage Waiting Time=%.3f \nAverage Turn Around Time=%.3f
\n",avgwt,avgtat);

      return 0;
}



//FIRST COME FIRST SERVE SCHEDULING ALGORITHM with arrival time
#include <stdio.h>
struct process{
      int pid;
```

```
      int bt;
      int wt,tt;
      int at;
}p[10],temp;

int main(){
      int n,totwt,tottt;
      float avgwt,avgtat;
      int i,j;

      printf("Enter the no of process \n");
      scanf("%d",&n);

      for(i=1;i<=n;i++){
            p[i].pid=i;
            printf("Enter the burst time for process %d\n",i);
            scanf("%d",&p[i].bt);
            printf("Enter the arrival time for process %d\n",i);
            scanf("%d",&p[i].at);
      }
      //sort according to arrival time
      for(i=1;i<=n;i++)
      {
            for(j=i+1;j<=n;j++)
            {
                  if(p[i].at > p[j].at)
                  {
                        temp=p[j];
                        p[j]=p[i];
                        p[i]=temp;
                  }
            }
      }

int totbt=0; //total burst time

      for(i=1;i<=n;i++)
      {
            if(i==1) // for first process
            {
                  p[i].wt=p[i].at;
                  p[i].tt=p[i].bt+p[i].wt;
                  totbt += p[i].bt;
            }
            else  //for rest of the process
            {
                  p[i].wt = totbt - p[i].at;
                  p[i].tt = p[i].bt+p[i].wt;
                  totbt += p[i].bt;
            }
      }
      printf("\n\nProcess Execution order is: \n");
      for(i=1;i<=n;i++)
            printf("P%d-->\t",p[i].pid);
```

```
       printf("\n\n");
       printf("\n Total Burst Time: %d\n",totbt);
      i=1;
      totwt=tottt=0;


      printf("\n Processid \t Burst Time\t Arrival Time\t Waiting Time \t
Turn Around Time\n");
      while(i<=n){
            printf("\n\t%d\t %d \t\t %d\t\t %d \t\t
%d",p[i].pid,p[i].bt,p[i].at,p[i].wt,p[i].tt);
            totwt=p[i].wt+totwt;
            tottt=p[i].tt+tottt;
            i++;
      }
      avgwt=(float)totwt/n;
      avgtat=(float)tottt/n;
      printf("\nAverage Waiting Time = %.3f \nAverage Turn Around Time =
%.3f \n",avgwt,avgtat);

      return 0;
}
```