```
/*
 * pre_sjf2.c
 *
 *  Created on:
 *      Author: root
 */
#include<stdio.h>
#include<string.h>
struct process
{
	char pname[10];
	int AT,BT,ST,FT,TT,WT,BT1;
}p[15];
struct process t;
int i,n,j,k,bt,tq;
char GC[200];
void get_data()
{
	printf("Enter number of processes : ");
	scanf("%d",&n);
	printf("Enter process details for %d processes",n);
	for(i=0;i<n;i++)
	{
		printf("\nEnter Process name,arival time,cpu burst time : ");
		scanf("%s   %d     %d",&p[i].pname,&p[i].AT,&p[i].BT);
		p[i].BT1=p[i].BT;
	}
}
void put_data()
{
	printf("\nProcesses are as below");
```

```
	printf("\nProcess name\t arival time\t cpu burst time");
	for(i=0;i<n;i++)
	{
		printf("\n%s\t\t%d\t\t%d",p[i].pname,p[i].AT,p[i].BT);
	}
}
void arrivalsort()
{
	//struct process t;
	for(i=0;i<n;i++)
	{
		for(j=i+1;j<n;j++)
		{
		   if(p[i].AT > p[j].AT)
		   {
			t=p[i];
			p[i]=p[j];
			p[j]=t;
		   }//if
		}//for
	}//for
}//arrivalsort

void burst_sort()
{
	//struct process t;
	for(i=0;i<n;i++)
	{
		for(j=i+1;j<n;j++)
		{
		   if(p[i].BT > p[j].BT)
```

```
                    {
                        t=p[i];
                        p[i]=p[j];
                        p[j]=t;
                    }//if
                }//for
        }//for
}//arrivalsort

void avgTTWT()
{
        float sumtt=0,sumwt=0;
        for(i=0;i<n;i++)
        {
                p[i].TT=p[i].FT-p[i].AT;
                p[i].WT=p[i].TT-p[i].AT;
                sumtt=sumtt+p[i].TT;
                sumwt=sumwt+p[i].WT;
        }//for
        printf("\n Process\tAT\tBT\tTT\tWT\n");
        for(i=0;i<n;i++)
                {

        printf("\n%s\t\t%d\t%d\t%d\t%d",p[i].pname,p[i].AT,p[i].BT1,p[i].TT,p[i].WT);
                }
                        printf("\nAverage turn around time =%f/%d = %f",sumtt,n,sumtt/n);
                        printf("\nAverage wait time=%f/%d = %f",sumwt,n,sumwt/n);
        }//avgTTWT
void pre_sjf()
{
        char str[5];
```

```
i=0;
int time=0;
tq=1;
            strcpy(GC,"0|");
            aaa:
            if(p[i].BT!=0)
            {
               if(p[i].AT>time)
               {
 for(j=i+1;j<n;j++)
 {
     if(p[j].AT < p[i].AT && p[j].BT!=0)
     {
                  time=p[j].AT;
                  sprintf(str,"%d",time);
                  strcat(GC,str);
                  strcat(GC,"|");
                  p[j].ST=time;
                                                strcat(GC,p[j].pname);
                                                p[j].BT=p[j].BT-tq;
                                                strcat(GC," ");
                                          time=time+tq;
                                                        sprintf(str,"%d",time);
                                                        strcat(GC,str);
                                                        p[j].FT=time;

      }//if
 }//for
                }//if
            /*else
            {
```

```
                        strcat(GC,"CPUIDLE");
                time=p[i].AT;
                        sprintf(str,"%d",time);
                        strcat(GC,str);
                        strcat(GC,"|");
        }*/

        p[i].ST=time;
        strcat(GC,p[i].pname);
        time=time+tq;
        strcat(GC," ");
    p[i].FT=time;
        sprintf(str,"%d",time);
        strcat(GC,str);
        strcat(GC,"|");
        p[i].BT=p[i].BT-tq;
        burst_sort();
        }//if
        for(i=0;i<n;i++)
        {
                if(p[i].BT!=0)
                        goto aaa;
        }//for
                printf("\nGantt Chart\n");
                puts(GC);
                avgTTWT();
}//rr

int main()
{
    get_data();
```

```
        put_data();
arrivalsort();
put_data();
pre_sjf();
}
/*
Enter number of processes : 3
Enter process details for 3 processes
Enter Process name,arival time,cpu burst time : P1 0 9

Enter Process name,arival time,cpu burst time : P2 1 5

Enter Process name,arival time,cpu burst time : P3 2 10

Processes are as below
Process name    arival time    cpu burst time
P1          0          9
P2          1          5
P3          2          10
Processes are as below
Process name    arival time    cpu burst time
P1          0          9
P2          1          5
P3          2          10
Gantt Chart
0|P1 1|P2 2|P2 3|P2 4|P2 5|P2 6|P1 7|P1 8|P1 9|P1 10|P1 11|P1 12|P1 13|P1 14|P3 15|P3 16|P3 17|P3 18|P3 19|P3 20|P3 21|P3 22|P3 23|P3 24|

 Process      AT     BT     TT     WT

P2          1      5      5      4
```

P1 0 9 14 14

P3 2 10 22 20

Average turn around time =41.000000/3 = 13.666667

Average wait time=38.000000/3 = 12.666667

*/