**Student Name:** Prince Paraste

**Student ID:** 11604723

**Email Address:** princeparaste78@gmail.com

**GitHub Link:** <https://github.com/prince214/OS_Project>

**Code:**

**#include<stdio.h>**

**#include<stdlib.h>**

**typedef struct process**

**{**

**int id,at,bt,st,ft,pr;**

**float wt,tat;**

**}process;**

**process p[100],p1[100],temp;**

**int accept(int ch);**

**void sort(int n);**

**void gantt(int n);**

**void turnwait(int n);**

**void display(int n);**

**int main()**

**{**

**int i,n,ts,ch,j,x;**

**p[0].tat=0;**

**p[0].wt=0;**

**n=accept(ch);**

**sort(n);**

**gantt(n);**

**turnwait(n);**

**display(n);**

**return 0;**

**}**

**int accept(int ch) //functions for accepting inputs from the user**

**{**

**int i,n;**

**printf("\nEnter the no. of process: ");**

**scanf("%d",&n);**

**if(n==0)**

**{**

**printf("\nInvalid no. of process");**

**exit(1);**

**}**

**for(i=1;i<=n;i++)**

**{**

**printf("\nEnter the arrival time for process P%d: ",i);**

**scanf("%d",&p[i].at);**

**p[i].id=i;**

**}**

**for(i=1;i<=n;i++)**

**{**

**printf("\nEnter the burst time for process P%d: ",i);**

**scanf("%d",&p[i].bt);**

**}**

**for(i=1;i<=n;i++)**

**{**

**printf("\nEnter the priority for P%d: ",i);**

**scanf("%d",&p[i].pr);**

**}**

**printf("\nInputs given by the user are:\n");**

**printf("==============================================================");**

**printf("\nProcess\tAT\tBT\tPR");**

**for(i=1;i<=n;i++)**

**printf("\nP%d\t%d\t%d\t%d",p[i].id,p[i].at,p[i].bt,p[i].pr);**

**printf("\n============================================================");**

**for(i=1;i<=n;i++) // copying of one array into another array(dummy area)**

**p1[i]=p[i];**

**return n;**

**}**

**void sort(int n)**

**{**

**int i,j;**

**for(i=2;i<=n;i++)**

**for(j=1;j<=i;j++)**

**if(p[j].at>p[i].at)**

**{**

**temp=p[i];**

**p[i]=p[j];**

**p[j]=temp;**

**}**

**}**

**/\* Function for displaying GANTT CHART FOR First Come First Serve Premptive Priority \*/**

**void gantt(int n)**

**{**

**int i,j,min,nextval,limit,ts,m=1;**

**limit=0;**

**nextval=p[1].at;**

**printf("\nEnter the time slice: ");**

**scanf("%d",&ts);**

**for(i=1;i<=n;i++)**

**limit=limit+p[i].bt;**

**limit=limit+p[1].at;**

**if(p1[1].bt<ts)**

**nextval=nextval+p1[1].bt;**

**else**

**nextval=nextval+ts;**

**printf("Gantt chart is as follows\n\n");**

**printf("%d",nextval);**

**if(p1[m].bt<ts)**

**p1[m].bt=0;**

**else**

**p1[m].bt=p1[m].bt-ts;**

**do**

**{**

**if(p1[m].bt<=0)**

**{**

**min=9999;**

**for(i=1;(p1[i].at)<=nextval,i<=n;i++)**

**if(p1[i].pr<min && p1[i].bt>0)**

**{**

**min=p1[i].pr;**

**m=i;**

**}**

**}**

**i=1;**

**while(i<=n)**

**{**

**if((p1[i].at)<=nextval)**

**if((p1[i].pr<p1[m].pr) && (p1[m].bt>0) && (p1[i].bt>0))**

**m=i;**

**i++;**

**}**

**if(p1[m].bt<ts)**

**nextval=nextval+p1[m].bt;**

**else**

**nextval=nextval+ts;**

**printf("->P%d->%d",p1[m].id,nextval);**

**if(p1[m].bt<ts)**

**p1[m].bt=0;**

**else**

**p1[m].bt=p1[m].bt-ts;**

**if(p1[m].bt==0)**

**{**

**for(i=1;i<=n;i++)**

**if(p[i].id==m)**

**{**

**p[i].ft=nextval;**

**}**

**}**

**}while(nextval<limit);**

**}**

**/\* FUNCTION FOR CALCULATING TURN AROUND TIME OR WAIT TIME \*/**

**void turnwait(int n)**

**{**

**int i;**

**for(i=1;i<=n;i++)**

**{**

**p[i].tat=p[i].ft-p[i].at;**

**p[i].wt=p[i].tat-p[i].bt;**

**p[0].tat=p[0].tat+p[i].tat;**

**p[0].wt=p[0].wt+p[i].wt;**

**}**

**p[0].tat=p[0].tat/n;**

**p[0].wt=p[0].wt/n;**

**}**

**/\* FUNCTION FOR DISPLAYING THE TABLE \*/**

**void display(int n)**

**{**

**int i;**

**printf("\n\n-------------------TABLE----------------------------------\n");**

**printf("\nProcess\tAT\tBT\tFT\tTAT\t\tWT");**

**for(i=1;i<=n;i++)**

**if(p[i].tat>=0 && p[i].wt>=0)**

**{**

**printf("\nP%d\t%d\t%d\t%d\t%f\t%f",p[i].id,p[i].at,p[i].bt,p[i].ft,p[i].tat,p[i].wt);**

**printf("\n\n-----------------------------------------------------------");**

**}**

**else**

**{**

**p[i].tat=0;**

**p[i].wt=0;**

**printf("\nP%d\t%d\t%d\t%d\t%f\t%f",p[i].id,p[i].at,p[i].bt,p[i].ft,p[i].tat,p[i].wt);**

**printf("\n\n-----------------------------------------------------------");**

**}**

**printf("\nAverage Turn Around Time: %f",p[0].tat);**

**printf("\nAverage Waiting Time: %f",p[0].wt);**

**}**

**Algorithm:**

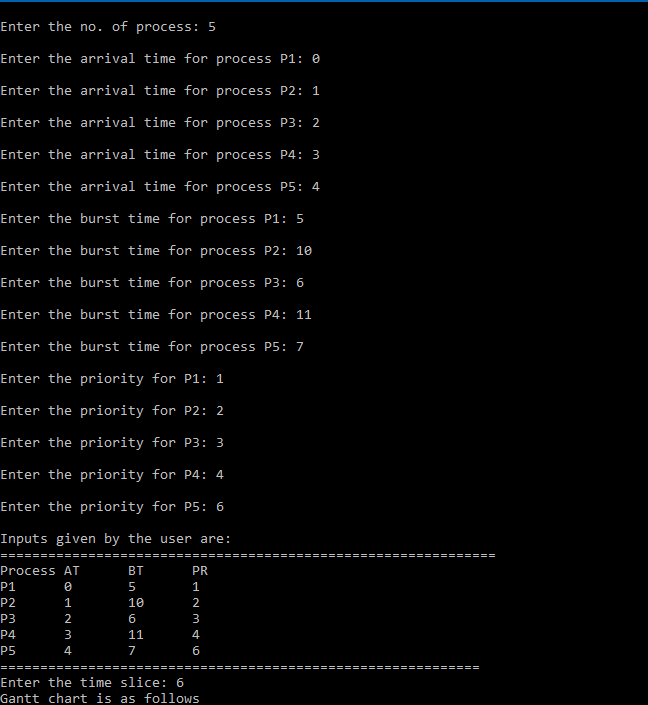
1. First input the processes with their burst time and priority.

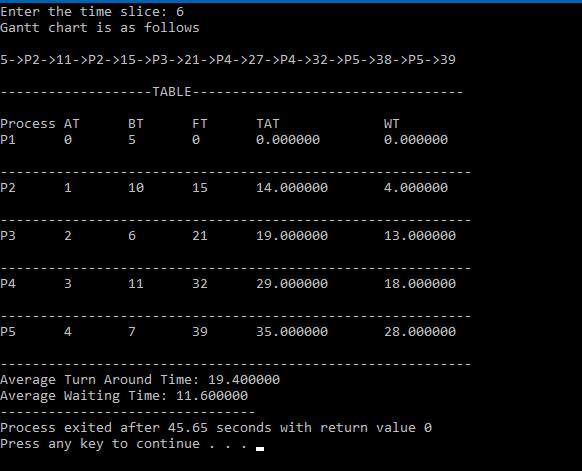
2. Sort the processes, burst time and priority according to the priority.

3. Now simply apply FCFS algorithm.

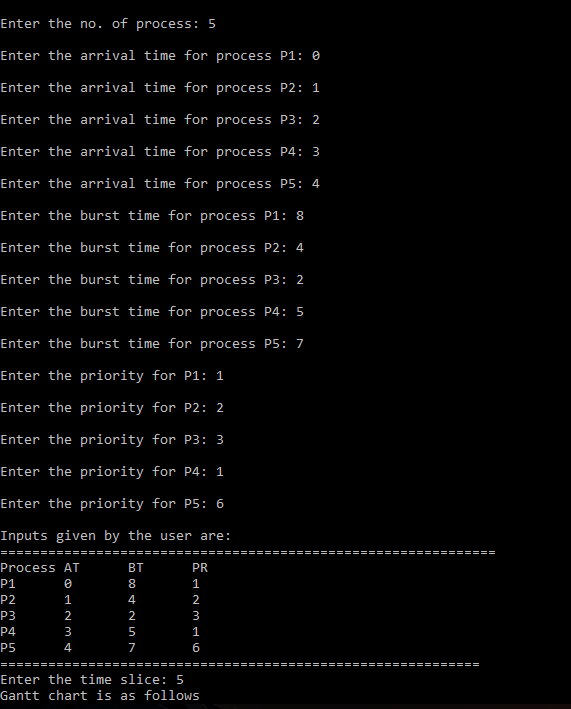
**Test Cases:**

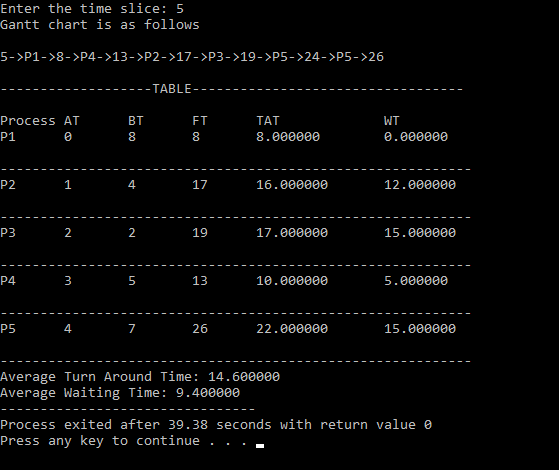
TestCase1:





TestCase2:





TestCase3:

