**LAB 2**

**Priority:**

#include<stdio.h>

void waitingtime(int proc[],int n,int bt\_time[],int wait\_time[]){

wait\_time[0]=0;

for(int i=1;i<n;i++){

wait\_time[i] = bt\_time[i-1] + wait\_time[i-1];

}

}

void turnaroundtime(int proc[],int n,int bt\_time[],int wait\_time[],int tat[]){

for(int i=0;i<n;i++){

tat[i] = bt\_time[i] + wait\_time[i];

}

}

int avgtime(int proc[], int n, int bt\_time[],int pr[]) {

int wait\_time[n], tat[n], total\_wt = 0, total\_tat = 0;

waitingtime(proc, n, bt\_time, wait\_time);

turnaroundtime(proc, n, bt\_time, wait\_time, tat);

printf("Process No.\tBurst Time\tWaiting Time\tTurnaround Time\tPriority\n");

for (int i = 0; i < n; i++) {

total\_wt = total\_wt + wait\_time[i];

total\_tat = total\_tat + tat[i];

printf("%d\t\t\t\t%d\t\t\t\t%d\t\t\t\t%d\t\t%d\n", i+1, bt\_time[i], wait\_time[i], tat[i],pr[i]);

}

printf("Average waiting time = %f\n", (float)total\_wt / (float)n);

printf("Average turn around time = %f\n", (float)total\_tat / (float)n);

}

int main() {

int i,j, temp, n,bt[10],proc[10],pr[10];

printf("Enter no of processes:");

scanf("%d",&n);

for(i=0;i<n;i++){

printf("Enter burst time of process %d:",i+1);

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

printf("Enter priority of process %d:",i+1);

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

proc[i]=i+1;

}

for(i=0;i<n;i++){

for(j=i+1;j<n;j++){

if(pr[i]>pr[j]){

temp=pr[i];

pr[i]=pr[j];

pr[j]=temp;

temp=proc[i];

proc[i]=proc[j];

proc[j]=temp;

temp=bt[i];

bt[i]=bt[j];

bt[j]=temp;

}

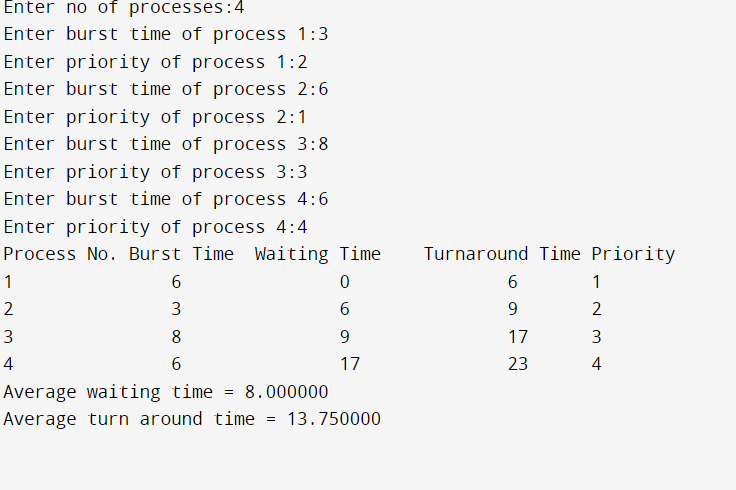
}

}

avgtime(proc, n, bt, pr);

return 0;

}



**Round Robin:**

#include<stdio.h>  
  
int main()  
{  
 int i, n, total = 0, x, counter = 0, time\_quantum;  
 int wait\_time = 0, turnaround\_time = 0, arrival\_time[10], burst\_time[10], temp[10];  
 float average\_wait\_time, average\_turnaround\_time;  
 printf("\nEnter Number of Processes:\t");  
 scanf("%d", &n);  
 x = n;  
 for(i = 0; i < n; i++)  
 {  
 printf("\nEnter Details of Process %d \n", i + 1);  
  
 printf("Arrival Time:\t");  
  
 scanf("%d", &arrival\_time[i]);  
  
 printf("Burst Time:\t");  
  
 scanf("%d", &burst\_time[i]);  
  
 temp[i] = burst\_time[i];  
 }  
  
 printf("\nEnter Time Quantum:\t");  
 scanf("%d", &time\_quantum);  
 printf("\nProcess No\tBurst Time\t Turnaround Time\t Waiting Time\n");  
 for(total = 0, i = 0; x != 0;)  
 {  
 if(temp[i] <= time\_quantum && temp[i] > 0)  
 {  
 total = total + temp[i];  
 temp[i] = 0;  
 counter = 1;  
 }  
 else if(temp[i] > 0)  
 {  
 temp[i] = temp[i] - time\_quantum;  
 total = total + time\_quantum;  
 }  
 if(temp[i] == 0 && counter == 1)  
 {  
 x--;  
 printf("\n%d\t\t\t%d\t\t\t %d\t\t\t %d", i + 1, burst\_time[i], total - arrival\_time[i], total - arrival\_time[i] - burst\_time[i]);  
 wait\_time = wait\_time + total - arrival\_time[i] - burst\_time[i];  
 turnaround\_time = turnaround\_time + total - arrival\_time[i];  
 counter = 0;  
 }  
 if(i == n - 1)  
 {  
 i = 0;  
 }  
 else if(arrival\_time[i + 1] <= total)  
 {  
 i++;  
 }  
 else  
 {  
 i = 0;  
 }  
 }  
  
 average\_wait\_time = wait\_time \* 1.0 / n;  
 average\_turnaround\_time = turnaround\_time \* 1.0 / n;  
 printf("\n\nAverage Waiting Time:\t%f", average\_wait\_time);  
 printf("\nAvg Turnaround Time:\t%f\n", average\_turnaround\_time);  
 return 0;  
}

