**Week – 2**

**Write a C program to implement the following CPU scheduling for the following algorithms**

**a)FCFS b)SJF c)Priority d)Round Robin**

**Code:**

#include<stdio.h>

int pri[100],choice,c[100],w[100],t[100],a[100],b[100],p[100],n;

void fcfssort(int n){

int s=0,i=0,j=0;

for(i=0;i<n-1;i++){

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

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

int t=a[i];

a[i]=a[j];

a[j]=t;

t=b[i];

b[i]=b[j];

b[j]=t;

t=p[i];

p[i]=p[j];

p[j]=t;

}

}

}

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

if(a[i]>=s) {

s=a[i];

}

s=s+b[i];

c[i]=s;

w[i]=c[i]-a[i]-b[i];

t[i]=b[i]+w[i];

}

}

void fcfs(int n){

int s=0,i=0;

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

if(a[i]>=s){

s=a[i];

}

s=s+b[i];

c[i]=s;

w[i]=c[i]-a[i]-b[i];

t[i]=b[i]+w[i];

}

}

void sort(int n){

int i=0,j=0;

for(i=0;i<n-1;i++){

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

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

int t=b[i];

b[i]=b[j];

b[j]=t;

t=a[i];

a[i]=a[j];

a[j]=t;

t=p[i];

p[i]=p[j];

p[j]=t;

t=pri[i];

pri[i]=pri[j];

pri[j]=t;

}

}

}

}

void sjf(int n){

sort(n);

fcfs(n);

display(n);

}

void prio(int n) {

int i = 0, j = 0;

for (i = 0; i < n - 1; i++) {

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

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

int t = pri[i];

pri[i] = pri[j];

pri[j] = t;

t = p[i];

p[i] = p[j];

p[j] = t;

t = a[i];

a[i] = a[j];

a[j] = t;

t = b[i];

b[i] = b[j];

b[j] = t;

}

}

}

fcfs(n);

display(n);

}

void robin(int n) {

int cnt, t, remain = n, flag = 0, tq;

int wt = 0, tat = 0, at[10], bt[10], rt[10];

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

at[i] = a[i];

bt[i] = rt[i] = b[i];

}

printf("Enter Time Quantum:\t");

scanf("%d", &tq);

printf("P\tAT\tBT\tCT\tWT\tTAT\n");

for (t = 0, cnt = 0; remain != 0;) {

if (rt[cnt] <= tq && rt[cnt] > 0) {

t += rt[cnt];

rt[cnt] = 0;

flag = 1;

} else if (rt[cnt] > 0) {

rt[cnt] -= tq;

t += tq;

}

if (rt[cnt] == 0 && flag == 1) {

remain--;

printf("%d\t%d\t%d\t%d\t%d\t%d\n", p[cnt], at[cnt], bt[cnt], t, t - at[cnt] - bt[cnt], t - at[cnt]);

wt += t - at[cnt] - bt[cnt];

tat += t - at[cnt];

flag = 0;

}

if (cnt == n - 1)

cnt = 0;

else if (at[cnt + 1] <= t)

cnt++;

else

cnt = 0;

}

printf("\nAverage Waiting Time= %f\n", wt \* 1.0 / n);

printf("Avg Turnaround Time = %f\n", tat \* 1.0 / n);

}

void display(int n){

int i=0;

float sw=0,st=0;

printf("P\tAT\tBT\tCT\tWT\tTAT");

if(choice == 3) {

printf("\tPriority");

}

printf("\n");

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

printf("%d\t%d\t%d\t%d\t%d\t%d",p[i],a[i],b[i],c[i],w[i],t[i]);

sw=sw+w[i];

st=st+t[i];

if(choice == 3){

printf("\t%d", pri[i]);

}

printf("\n");

}

printf("Average waiting time %f\n",sw/n);

printf("Average turn around time %f\n",st/n);

}

int main(){

int i=0;

abc:

printf("Choose Options -> 1:FCFS 2:SJF 3:Priority 4:Round Robin 5:Exit\n");

printf("Enter choice: ");

scanf("%d",&choice);

printf("Enter no of processes\n");

scanf("%d",&n);

printf("\nEnter arrival times\n");

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

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

}

printf("\nEnter burst times\n");

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

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

}

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

p[i]=i;

pri[i]=0;

}

switch(choice){

case 1:

fcfssort(n);

display(n);

goto abc;

break;

case 2:

sjf(n);

goto abc;

break;

case 3:

printf("Enter priority ");

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

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

}

prio(n);

goto abc;

break;

case 4:robin(n);

goto abc;

break;

default:

break;

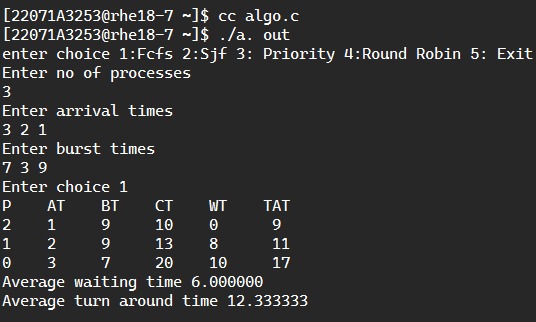
}

return 0;

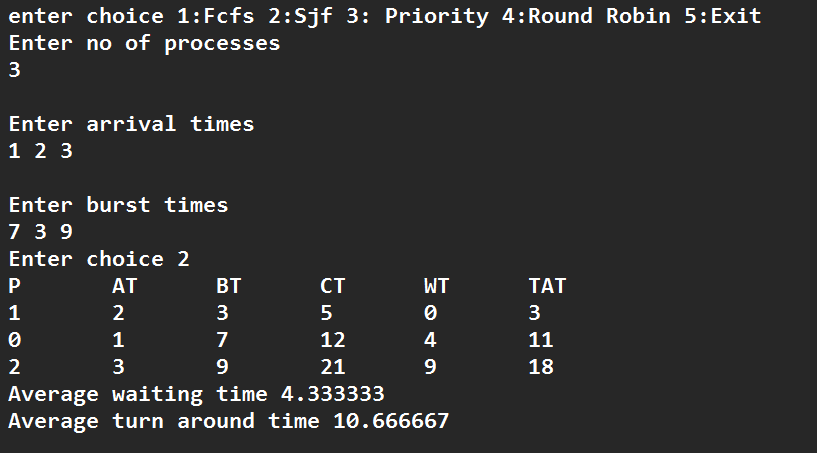
}

**Outputs:**

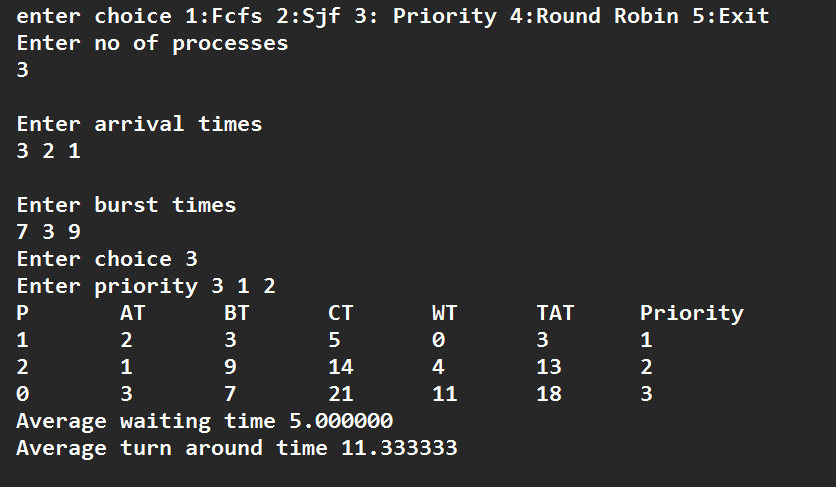
**FCFS Output:**



**SJF Output:**

****

**Priority Output:**

****

**Round Robin Output:**

