**EXP 3.** Write a program to simulate CPU Scheduling Algorithms: FCFS, SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive).

1. **FCFS Program**

**Main.java**

import java.util.\*;

public class Main

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

System.out.println("enter no of process: ");

int n = sc.nextInt();

int pid[] = new int[n]; // process ids

int ar[] = new int[n]; // arrival times

int bt[] = new int[n]; // burst or execution times

int ct[] = new int[n]; // completion times

int ta[] = new int[n]; // turn around times

int wt[] = new int[n]; // waiting times

int temp;

float avgwt=0,avgta=0;

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

{

System.out.println("enter process " + (i+1) + " arrival time: ");

ar[i] = sc.nextInt();

System.out.println("enter process " + (i+1) + " brust time: ");

bt[i] = sc.nextInt();

pid[i] = i+1;

}

//sorting according to arrival times

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

{

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

{

if( ar[j] > ar[j+1] )

{

temp = ar[j];

ar[j] = ar[j+1];

ar[j+1] = temp;

temp = bt[j];

bt[j] = bt[j+1];

bt[j+1] = temp;

temp = pid[j];

pid[j] = pid[j+1];

pid[j+1] = temp;

}

}

}

// finding completion times

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

{

if( i == 0)

{

ct[i] = ar[i] + bt[i];

}

else

{

if( ar[i] > ct[i-1])

{

ct[i] = ar[i] + bt[i];

}

else

ct[i] = ct[i-1] + bt[i];

}

ta[i] = ct[i] - ar[i] ; // turnaround time= completion time- arrival time

wt[i] = ta[i] - bt[i] ; // waiting time= turnaround time- burst time

avgwt += wt[i] ; // total waiting time

avgta += ta[i] ; // total turnaround time

}

System.out.println("\npid arrival brust complete turn waiting");

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

{

System.out.println(pid[i] + " \t " + ar[i] + "\t" + bt[i] + "\t" + ct[i] + "\t" + ta[i] + "\t" + wt[i] ) ;

}

sc.close();

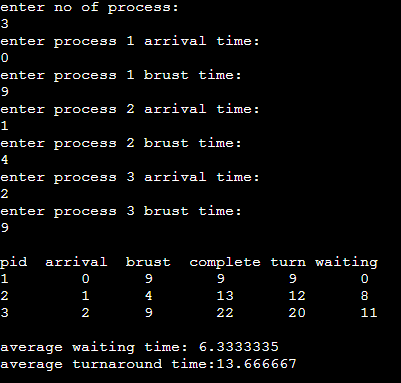
System.out.println("\naverage waiting time: "+ (avgwt/n)); // printing average waiting time.

System.out.println("average turnaround time:"+(avgta/n)); // printing average turnaround time.

}

}

**Output**



1. **SJF Program**

**Main.java**

import java.util.\*;

class Main {

static int[][] mat = new int[10][6];

static void arrangeArrival(int num, int[][] mat)

{

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

for (int j = 0; j < num - i - 1; j++) {

if (mat[j][1] > mat[j + 1][1]) {

for (int k = 0; k < 5; k++) {

int temp = mat[j][k];

mat[j][k] = mat[j + 1][k];

mat[j + 1][k] = temp;

}

}

}

}

}

static void completionTime(int num, int[][] mat)

{

int temp, val = -1;

mat[0][3] = mat[0][1] + mat[0][2];

mat[0][5] = mat[0][3] - mat[0][1];

mat[0][4] = mat[0][5] - mat[0][2];

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

temp = mat[i - 1][3];

int low = mat[i][2];

for (int j = i; j < num; j++) {

if (temp >= mat[j][1] && low >= mat[j][2]) {

low = mat[j][2];

val = j;

}

}

mat[val][3] = temp + mat[val][2];

mat[val][5] = mat[val][3] - mat[val][1];

mat[val][4] = mat[val][5] - mat[val][2];

for (int k = 0; k < 6; k++) {

int tem = mat[val][k];

mat[val][k] = mat[i][k];

mat[i][k] = tem;

}

}

}

// Driver Code

public static void main(String[] args)

{

int num;

Scanner sc = new Scanner(System.in);

System.out.println("Enter number of Process: ");

num = sc.nextInt();

System.out.println("...Enter the process ID...");

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

System.out.println("...Process " + (i + 1)

+ "...");

System.out.println("Enter Process Id: ");

mat[i][0] = sc.nextInt();

System.out.println("Enter Arrival Time: ");

mat[i][1] = sc.nextInt();

System.out.println("Enter Burst Time: ");

mat[i][2] = sc.nextInt();

}

System.out.println("Before Arrange...");

System.out.println(

"Process ID\tArrival Time\tBurst Time");

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

System.out.printf("%d\t\t%d\t\t%d\n", mat[i][0],

mat[i][1], mat[i][2]);

}

arrangeArrival(num, mat);

completionTime(num, mat);

System.out.println("Final Result...");

System.out.println(

"Process ID\tArrival Time\tBurst"

+ " Time\tWaiting Time\tTurnaround Time");

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

System.out.printf(

"%d\t\t%d\t\t%d\t\t%d\t\t%d\n", mat[i][0],

mat[i][1], mat[i][2], mat[i][4], mat[i][5]);

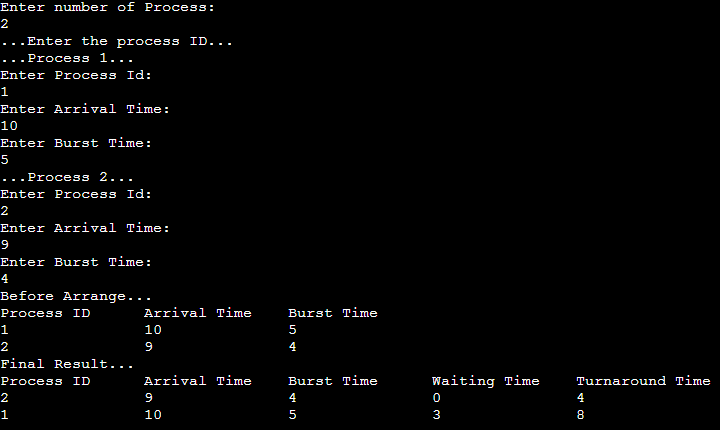
}

sc.close();

}

}

**Output**



1. **Priority Program**

**Main.java**

import java.util.Scanner;

public class Main{

public static void main(String args[]) {

Scanner s = new Scanner(System.in);

int x,n,p[],pp[],bt[],w[],t[],awt,atat,i;

p = new int[10];

pp = new int[10];

bt = new int[10];

w = new int[10];

t = new int[10];

//n is number of process

//p is process

//pp is process priority

//bt is process burst time

//w is wait time

// t is turnaround time

//awt is average waiting time

//atat is average turnaround time

System.out.print("Enter the number of process : ");

n = s.nextInt();

System.out.print("\n\t Enter burst time : time priorities \n");

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

{

System.out.print("\nProcess["+(i+1)+"]:");

bt[i] = s.nextInt();

pp[i] = s.nextInt();

p[i]=i+1;

}

//sorting on the basis of priority

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

{

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

{

if(pp[i]>pp[j])

{

x=pp[i];

pp[i]=pp[j];

pp[j]=x;

x=bt[i];

bt[i]=bt[j];

bt[j]=x;

x=p[i];

p[i]=p[j];

p[j]=x;

}

}

}

w[0]=0;

awt=0;

t[0]=bt[0];

atat=t[0];

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

{

w[i]=t[i-1];

awt+=w[i];

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

atat+=t[i];

}

//Displaying the process

System.out.print("\n\nProcess \t Burst Time \t Wait Time \t Turn Around Time Priority \n");

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

System.out.print("\n "+p[i]+"\t\t "+bt[i]+"\t\t "+w[i]+"\t\t "+t[i]+"\t\t "+pp[i]+"\n");

awt/=n;

atat/=n;

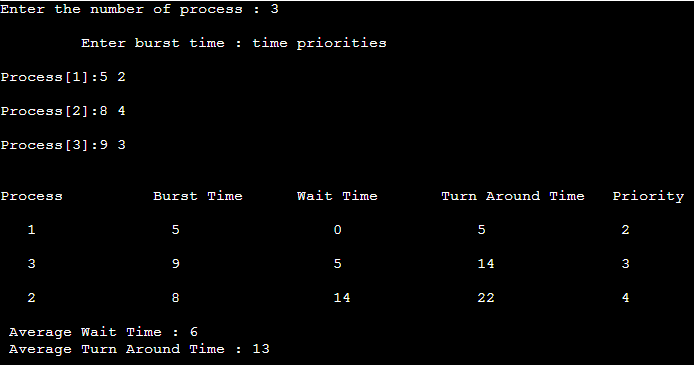
System.out.print("\n Average Wait Time : "+awt);

System.out.print("\n Average Turn Around Time : "+atat);

}

}

**Output**



1. **Round Robin**

**Main.java**

import java.io.\*;

class Main{

public static void main(String args[])throws IOException{

DataInputStream in=new DataInputStream(System.in);

int i,j,k,q,sum=0;

System.out.println("Enter number of process:");

int n=Integer.parseInt(in.readLine());

int bt[]=new int[n];

int wt[]=new int[n];

int tat[]=new int[n];

int a[]=new int[n];

System.out.println("Enter brust Time:");

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

System.out.println("Enter brust Time for "+(i+1));

bt[i]=Integer.parseInt(in.readLine());

}

System.out.println("Enter Time quantum:");

q=Integer.parseInt(in.readLine());

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

a[i]=bt[i];

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

wt[i]=0;

do{

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

if(bt[i]>q){

bt[i]-=q;

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

if((j!=i)&&(bt[j]!=0))

wt[j]+=q;

}

}

else{

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

if((j!=i)&&(bt[j]!=0))

wt[j]+=bt[i];

}

bt[i]=0;

}

}

sum=0;

for(k=0;k<n;k++)

sum=sum+bt[k];

}

while(sum!=0);

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

tat[i]=wt[i]+a[i];

System.out.println("process\t\tBT\tWT\tTAT");

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

System.out.println("process"+(i+1)+"\t"+a[i]+"\t"+wt[i]+"\t"+tat[i]);

}

float avg\_wt=0;

float avg\_tat=0;

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

avg\_wt+=wt[j];

}

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

avg\_tat+=tat[j];

}

System.out.println("average waiting time "+(avg\_wt/n)+"\n Average turn around time"+(avg\_tat/n));

}

}

**Output**

