# ROUND ROBBIN CODE

```
import java.util.*;
import java.io.*;

public class robbin
{

    public static void main(String args[])
    {
        int n,sum=0;
        float total_tt=0,total_waiting=0;

          Scanner s=new Scanner(System.in);
          System.out.println("Enter Number Of Process U want 2 Execute---");
          n=s.nextInt();
          int arrival[]=new int[n];
          int cpu[]=new int[n];
          int ncpu[]=new int[n];
          int pri[]=new int[n];
          int finish[]=new int[100];
          int turntt[]=new int[n];
          int wait[]=new int[n];
          int process[]=new int[n];
          int t_quantum,difference,temp_sum=0,k=0;
          int seq[]=new int[100];

         // int pro[][]=new int[3][3];
          for(int i=0;i<n;i++)
          {
```

```
      System.out.println("Enter arrival time of "+(i+1)+" Process : ");
      arrival[i]=s.nextInt();
      System.out.println("Enter CPU time of "+(i+1)+" Process : ");
      ncpu[i]=cpu[i]=s.nextInt();



      process[i]=i+1;
 }

System.out.println("Enter time quantum : ");
t_quantum = s.nextInt();

 int tv=0;
 for(int i=0;i<n;i++){temp_sum=temp_sum+cpu[i];}
 //System.out.println(temp_sum);


 System.out.println("Process execution sequence : ");
 while(sum!=temp_sum){
 for(int i=0;i<n;i++)
 {
  if(ncpu[i]<t_quantum)
      {
         difference=ncpu[i];
         tv=ncpu[i];
         ncpu[i]=0;
      }
  else
```

```
      {
          difference = ncpu[i]-t_quantum;
          tv=t_quantum;
          ncpu[i]=difference;
      }
  if(tv > 0)
  {
      sum=sum+tv;
      finish[k]=sum;
      seq[k]=i;
      System.out.print(seq[k]+1+" ");

      k++;


      }
}
}
System.out.println();

for(int i=0;i<n;i++)
{
  int carr=0,tt=0;
  carr=arrival[i];

  for(int j=0;j<k;j++)
  {
          if(seq[j]==i)
          {
              tt=tt+(finish[j]-carr);
```

```
                carr=finish[j];
            }
        }

        turntt[i]=tt;
        System.out.println("Turn around time for "+(i+1)+" process : "+turntt[i]);
        total_tt=total_tt+turntt[i];

        wait[i]=turntt[i]-cpu[i];

        System.out.println("Waiting time for "+(i+1)+" process : "+wait[i]);

        total_waiting+=wait[i];
    }

    System.out.println("\n\nProcess\t\tAT\tCPU_T");
    for(int i=0;i<n;i++)
    {
        System.out.println(process[i]+"\t\t"+arrival[i]+"\t"+cpu[i]);
    }

    System.out.println("\n\n");
    System.out.println("Total turn around time is : "+(total_tt/n));
    System.out.println("Total waiting time is : "+(total_waiting/n));


  }
}
```

# OUTPUT

```
Enter Number Of Process U want 2 Execute---3

Enter arrival time of 1 Process :

0

Enter CPU time of 1 Process :

3

Enter arrival time of 2 Process :

1

Enter CPU time of 2 Process :

4

Enter arrival time of 3 Process :

4

Enter CPU time of 3 Process :

6

Enter time quantum : 2

Process execution sequence :

1 2 3 1 2 3 3

Turn around time for 1 process : 7

Waiting time for 1 process : 4

Turn around time for 2 process : 8

Waiting time for 2 process : 4

Turn around time for 3 process : 9

Waiting time for 3 process : 3

Process      AT     CPU_T

1          0     3

2          1     4

3          4     6

Total turn around time is : 8.0

Total waiting time is : 3.6666667
```