

```

#include<stdio.h>
#define max 100
//queue essentials
int queue[max];
int front=-1, rear=-1; //making front 0 for avoiding much conditions while
dequeue

int pid[4]={1,2,3,4};
int a[4]={0,1,2,3};
int b[4]={8,5,10,11};
int rbt[4]={8,5,10,11};
int sts[4]={0,0,0,0};
int ct[4]={0,0,0,0};
int attendance[]={0,0,0,0};
int count=4;
int jtime=0;
int tq=2;

void enqueue(int item)
{
    if(rear==max-1)
    {
        printf("Queue full \n");
    }
    else
    {
        if(front== -1)
        {
            front=0;
        }
        rear++;
        queue[rear]=item;
        attendance[item]=1;
    }
}

int dequeue()
{
    int x=-1;
    if(front>rear)
    {
        printf("Queue empty \n");
    }
    else
    {
        x=queue[front];
        front++;
        attendance[x]=0;
    }
    return x;
}

void add_by_time(int time)

```

```

{
    for(int i=0;i<4;i++)
    {
        if(a[i]<=time&&sts[i]!=1&&attendance[i]!=1)
        {
            enqueue(i);
        }
    }
}

```

```

void process_q()
{
    int i=dequeue();
    attendance[i]=1;

    if(rbt[i]<=tq&&rbt[i]>0)
    {
        jtime=jtime+rbt[i];
        ct[i]=jtime;
        rbt[i]=0;
        sts[i]=1;
        count--;
    }
    else if(rbt[i]>0)
    {
        rbt[i]=rbt[i]-tq;
        jtime=jtime+tq;
    }

    add_by_time(jtime);

    attendance[i]=0;

    if(sts[i]!=1)
    {
        enqueue(i);
    }
}

```

```

int main()
{
    add_by_time(0); //adding the initial time ready process to ready q

```

```

//calculating the completion time

```

```

while(count!=0)
{

    process_q();

}

//Printing the results

printf("Process      Arrival time      Burst time      completion time
\n");

for(int i=0;i<4;i++)
{
    printf("      %d      %d      %d
%d\n",pid[i],a[i],b[i],ct[i]);
}
}

```