

```

#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]={0,1,2,4};
int b[4]={5,3,4,1};
int rbt[4]={5,3,4,1};
int sts[4]={0,0,0,0};
int ct[4]={0,0,0,0};
int count=4;
int jtime=100; //just initialized to high

```

```

void process_Ps()
{
    int index=-1;
    int min_rbt=100;

    for(int i=0;i<4;i++)
    {
        if(a[i]<=jtime&&sts[i]!=1)
        {
            if(rbt[i]<min_rbt)
            {
                min_rbt=rbt[i];
                index=i;
            }
        }
    }
}

```

```

//got the index
if(index!=-1)
{
    if(rbt[index]>0)
    {
        rbt[index]--;
        jtime++;
    }
    //if process is completed
    if(rbt[index]==0)
    {
        ct[index]=jtime;
        sts[index]=1;
        count--;
    }
}
}

```

```

        else //else if index not found
        {
            jtime++;
        }
    }

int main()
{
    //finding the process having least arrival time

    for(int i=0;i<4;i++)
    {
        if(a[i]<jtime)
        {
            jtime=a[i];
        }
    }

    //calculating the completion time

    while(count!=0)
    {

        process_Ps();

    }

    //Printing the results

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

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

```