#include<stdio.h>

#include<unistd.h>

#include<stdlib.h>

#include<conio.h>

#include<stdbool.h>

struct process{

int id;

int Arrivaltime;

int Bursttime;

int priority;

int status;

}pro[10];

bool compare(struct process a, struct process b)

{

return a.Arrivaltime < b.Arrivaltime;

}

void main()

{

int n,i,j;

int tq = 10;

int time = 0;

int count = 0, count2 = 0;

printf("Number of processes:\n");

scanf("%d",&n);

printf("Arrival time of each process:\n");

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

{

printf("Arrival time of P%d:\n",n+1);

scanf("%d",&pro[i].Arrivaltime);

pro[i].id = i+1;

}

printf("Burst time of each process:\n");

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

{

printf("Burst time of P%d:",i+1);

scanf("%d",&pro[i].Bursttime);

}

printf("Priority of each process:\n");

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

{

printf("Priority of P%d:\n",i+1);

scanf("%d",&pro[i].priority);

}

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

pro[i].status = 0;

qsort(pro,n,sizeof(pro[0]), compare);

i=0;

while(count!=n)

{

if(pro[i].status!=1 && pro[i].status!=2)

{

printf("Processing %d\n",pro[i].id);

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

{

if(pro[j].status != 1 && pro[j].status!=2 && (pro[j].priority < pro[i].priority) && (pro[j].Arrivaltime <= time))

{

printf("Process P%d is interuppted by P%d.Pushing it into queue2\n",pro[i].id,pro[j].id);

printf("Process P%d is processing",pro[j].id);

pro[i].status = 2;

count2++;

time += pro[j].Bursttime;

pro[j].status = 1;

printf("Process P%d is completely processed\n",pro[j].id);

count += 2;

break;

}

}

if(pro[i].status != 2)

{

time += pro[i].Bursttime;

pro[i].status = 1;

printf("Process P%d is completely processed",pro[i].id);

count++;

}

}

i = (i+1)%n;

}

time = 0;

count = 0;

printf("Executing Queue2\n");

i=0;

while(count != count2)

{

if(pro[i].status == 2)

{

if(pro[i].Bursttime<10)

{

time +=pro[i].Bursttime;

pro[i].Bursttime =0;

pro[i].status = 1;

printf("Completed P%d\n",pro[i].id);

count++;

}

else

{

pro[i].Bursttime -= 10;

time += tq;

printf("Processing P%d\n",pro[i].id);

}

}

i = (i+1)%n;

}

}