**题目1：进程调度1—静态非剥夺式优先级调度计算平均作业周转时间**

问题描述：要求输入3个进程的信息，假设这些进程均是在0时刻同时到达，若进程调度采用非剥夺式静态优先级（优先数数值大的表示优先级比较高；如果遇到优先级一样，按照输入顺序执行。），计算并输出平均作业周转时间。

输入格式：程序要求输入3行，以回车符号作为分隔，每行有3个数据，以空格作为分隔。首先输入一个字符串（长度小于等于10），为进程名，第2个数据类型为整型，表示进程的优先数，第3个数据类型为整型，表示进程的运行时间。

输出格式：输出结果为一个浮点数，保留到小数点后一位，为系统的平均作业周转时间。

样例输入1：

P1 1 1

P2 2 2

P3 3 3

样例输出1：

4.7

**题目2：进程调度2--最高响应比优先计算每个作业的周转时间**

问题描述：要求输入3个进程的信息，按照最高响应比优先的调度算法计算并输出每个进程的周转时间。(若两个进程的响应比相同，则优先选择先进入的进程。若两个进程的响应比相同，而且进入时刻也相同，则按照输入的顺序执行，如：P4和P6的响应比相同且进入时刻也相同，如P4先输入则选择P4先执行)

输入格式：程序要求输入3行，以回车符号作为分隔，每行有3个数据，以空格作为分隔。首先输入一个字符串（长度小于等于10），为进程名，第2个数据类型为整型，表示进程的进入时刻，第3个数据类型为整型，表示进程的运行时间。

输出格式：输出三个整数之间，整数之间用空格作为分隔，为每个进程的周转时间。

样例输入1：

P1 1 1

P2 2 2

P3 3 3

样例输出1：

1 2 4

**题目3：进程调度3**

问题描述：要求输入N个进程（N为正整型数，0<N<=25535），输出按照优先级从高到低执行的进程名字符串序列,直至结束。（如果遇到优先级一样，按照输入顺序先后执行。），本题中，优先数数值较高的进程，优先级也较高。

输入格式：程序首先要求输入一个整型变量N，接下来输入为N行，以回车符号作为分隔，每行有3个数据，以空格作为分隔。首先输入一个字符串（长度小于等于10），该字符串为进程名。第2个数据类型为整型，表示进程的优先数。第3个数据类型为整型，表示进程的运行时间。

输出格式：输出1行，M个字符串，字符串之间用空格作为分隔。

样例输入1：

3

P1 1 1

P2 2 2

P3 3 3

样例输出1：

P3 P2 P3 P1 P2 P3

**题目4：进程调度4----时间片轮转**

问题描述：要求输入N个进程（0<N<=100），输入时间片M（0<M〈=5），按照进程输入的顺序以时间片轮转的方法输出指定的第K轮（K>0）执行的那个进程的进程名。

输入格式：程序首先输入一个正整数M（0<M〈=5）作为时间片，下一行输入一个正整数N（0<N<=100），接下来输入为N行，以回车符号作为分隔，每行有2个数据，以空格作为分隔。第一个数据是字符串（长度小于等于10），该字符串为进程名，第2个数据类型为整型，表示该进程需要的运行时间。最后输入一个正整数K，作为时间片轮转的次数（次数从1开始计数）。

输出格式：输出一个字符串，为最后执行进程的进程名；若无进程运行，则输出“over”（不含双引号，所有字母皆为小写）。

样例输入1：

1

3

P1 1

P2 2

P3 3

3

样例输出1：P3

**题目5：死锁—利用银行家算法判断系统的安全性**

问题描述：假设系统中有A、B、C三类资源，且有四个并发进程，要求输入资源总量Resource，以及每个进程运行所需的资源总量Claim和已经分配得到的资源量Allocation，利用银行家算法判断当前状态是否为安全状态，若为安全状态则给出一个安全序列。

输入格式：程序要求输入五行，以回车符号作为分隔。第一行是三个整数，整数之间以空格作为分隔，表示A、B、C三类资源的总量。下面的四行分别表示每个进程运行所需的资源总量Claim和已经分配得到的资源量Allocation；每行有7个数据，以空格作为分隔。首先输入一个字符串（长度小于等于10），为进程名；第2、3、4个数据类型为整型，表示相应进程运行所需A、B、C三种资源总量Claim；第5、6、7个数据类型为整型，表示相应进程已经分配得到的A、B、C三种资源量Allocation。

输出格式：输出一个字符串。若当前为不安全状态则输出为”false”（不含双引号，所有字母皆为小写）；若当前为安全状态则输出一个安全序列，进程名之间用空格作为分隔）。

样例输入1：

9 5 7

P1 5 3 4 2 1 3

P2 9 5 2 2 1 1

P3 3 2 2 2 2 1

P4 6 4 1 1 1 1

样例输出1：

P3 P1 P4 P2

**题目6：存储管理—可变分区存储管理方式的最佳适应分配算法**

问题描述：当内存管理采用可变分区分配方案时，要求输入多个空闲分区和进程内存请求序列，输出显示采用最佳适应分配算法分配给各个进程的分区编号。

输入格式：程序要求输入3行，以回车符号作为分隔，第一行是一个整数n（n>=3），表示空闲分区的数量；第二行是n个整数，依次按地址递增对应第一行n个空闲分区的存储容量，每个整型数的数值代表所对应空间的剩余存储容量。n个分区（分区按地址递增依次从1开始编号，若分区X被切割分配了，剩余部分即使为0也保留原来的分区编号X）。第三行是3个整数，两个数之间以空格作为分隔，分别表示三个进程先后依次申请的内存空间的大小。

输出格式：输出一行三个整数，整数之间用空格作为分隔，分别表示三个进程所分配的分区编号；若分配失败，则用”false”表示（不含双引号，所有字母皆为小写）。

样例输入1：

6

20 5 6 18 60 4

12 7 20

样例输出1：

4 1 5

**题目7：存储管理—FIFO页面替换算法计算中断次数**

问题描述：在请求分页式存储管理方式中，要求输入一个对5个页面的访问序列，输出当系统分配给进程物理页框数为m个时，按照FIFO页面替换算法的缺页中断次数（假设初始时页框均为空）。

输入格式：程序要求输入3行，以回车符号作为分隔，第一行是一个整数n，表示页面访问序列中访问页面的次数；第二行是n个整数，数之间以空格作为分隔，表示页面访问序列。第三行是一个整数m，表示系统分配给进程物理页框数。

输出格式：输出一个整数，表示缺页中断次数。

样例输入1：

12

4 3 2 1 4 3 5 4 3 2 1 5

3

样例输出1：

9

**题目8：存储管理1**

问题描述：现有一个8\*8的存储器，要对其空间进行分配。（下标从0开始，最后一个内存块下标为63）。现已有块号为1、7、13、23、47、59的几个内存块被占用。现操作系统要求申请N块内存空间（0<N<=64），当输入的块数N超出其剩余空闲块数的时候，输出为“false”，当输入为合理范围的时候，就输出其以行主序分配的最后一个内存空间的下标。

输入格式：程序要求输入一个整型数N，表示要申请分配空间的大小。

输出格式：输出为一个整型数，表示最后一个被分配空间的下标。

样例输入1：

3

样例输出1：

3

**题目9：存储管理2**

问题描述：现有一个8\*8的存储器，要对其空间进行分配。（下标从0开始，最后一个内存块下标为63）。现已有块号为2、7、13、23、37、47、59、61的几个内存块被占用。要求输入需分配的进程数M（0<M<=56），接下来输入为M个整型数，每个数为各个进程需占用的内存块数。当分配到某进程时，其剩余空闲块数可以分配，就输出当前进程分配的最后一个内存空间的下标。当分配到某进程时，其进程块数超出剩余空闲块数无法分配，输出为“false”（不含双引号，且为全小写）。输出的多个下标（或"false"）之间用空格隔开。

输入格式：程序输入分为两行，第一行要求输入一个整型数M，表示要所需分配空间的进程数，接下来的第二行输入M个整型数，每个数之间用空格隔开，表示M个进程每个进程占用的内存空间大小。

输出格式：输出为M组整型数（或"false"），每个整型数表示该进程最后一个被分配的内存空间的下标（或"false"），下标（或"false"）之间用空格隔开。

样例输入1：

3

3 3 3

样例输出1：

3 6 10

**题目10：存储管理3**

问题描述：现有一个8\*8的存储器，要对其已分配的空间进行分配及回收。（下标从0开始，最后一个内存块下标为63）。现已有块号为2、7、13、23、37、41、47、59、61的几个内存块被占用。要求输入需分配的进程数M（0<M<=55），接下来输入为M个整型数，每个数为各个进程需占用的内存块数。当分配到某进程时，其剩余空闲块数可以分配，就输出当前进程分配的最后一个内存空间的下标。当分配到某进程时，其进程块数超出剩余空闲块数无法分配，输出为“false”（不含双引号，且为全小写）。输出的多个下标（或"false"）之间用空格隔开。以上进程不管是否分配成功，按照输入顺序依次命名为p1、p2、p3………pM。回收的时候输入进程名pN，则返回进程名为pN的所有占用内存块号下标，如果该进程名不存在或输入的数值为不合理范围，则返回“false”。

输入格式：程序输入分为三行，第一行是一个整型数M，表示要所需分配空间的进程数，第二行为M个整型数，每个数之间用空格隔开，表示M个进程每个进程占用的内存空间大小。第三行为需要回收的进程名pN，p为小写字母，N为正整型数。

输出格式：输出为两行，第一行为一组整型数，每个整型数表示该进程最后一个被分配的内存空间的下标，下标之间用空格隔开。第二行为一组整型数，表示被回收的进程的内存块下标，多个下标之间用空格隔开。

样例输入1：

3

3 3 3

p3

样例输出1：

3 6 10

8 9 10

**题目11：带存储管理的处理器调度4**

问题描述：现有一个内存为100K的采用位示图进行页面管理的道数不受限制的多道程序设计系统，若作业调度采用高优先级（优先数越大优先级越大）调度算法（如果遇到优先级一样且只能调入一道作业时，按照输入顺序选择调度对象。），进程调度采用非剥夺式的SJF调度算法（如果遇到运行时间相同的进程，按照输入顺序选择调度对象。）。要求输入3个进程信息，输出当三个作业同时提交进入调度时进程的运行顺序。

输入格式：程序要求输入3行，以回车符号作为分隔，每行有4个数据，以空格作为分隔。首先输入一个字符串（长度小于等于10），为进程名；第2个数据类型为整型，表示进程所需的内存空间；第3个数据类型为整型，表示进程的运行时间；第4个数据类型为整型，表示进程的优先数。

输出格式：输出1行，M个字符串，字符串之间用空格作为分隔。

样例输入1：

P1 20 2 1

P2 60 3 2

P3 30 4 3

样例输出1：

P2 P1 P3

**题目12：驱动调度—采用电梯调度算法排列出磁盘请求响应次序**

问题描述：要求输入一个柱面访问请求序列以及当前磁头所在柱面号和移动方向，输出采用电梯调度算法时移动臂响应的柱面访问序列。

输入格式：程序要求输入3行，以回车符号作为分隔，第一行是2个整数n、m，之间用空格隔开，n表示当前磁头所在的柱面号；m表示第二行输入m个数；第二行是m个整数，数之间以空格作为分隔，表示柱面访问请求序列；第三行是数字-1或1，当为-1时表示移动臂向柱面号减小方向移动，当为1时表示移动臂向柱面号增大方向移动。

输出格式：输出m个整数，数之间以空格作为分隔，采用电梯调度算法时移动臂响应的柱面访问序列。

样例输入1：

15 10

24 38 2 110 43 36 5 11 6 180

-1

样例输出1：

11 6 5 2 24 36 38 43 110 180