**《数据结构》上机实验题目**

**（共8次，每次上机4小时）**

**第一阶段（线性部分）**

**《数据结构》第1次上机题 （线性表练习）**

1．编程实现书P12 ADT List 基本操作13个：

* 1. 用顺序存储结构实现； （2）用链式存储结构实现；

1. 设元素值为整型的线性表L，分别采用顺序结构和链式结构存储，编写函数，实现线性表的就地逆置（书P31 4）。
2. 设线性表L，元素值为整型的且存在相同值，分别采用顺序结构和链式结构存储，编写函数，利用原空间，删除重复的元素值。
3. CSP 题目

最近西西艾弗岛上出入各个场所都要持有一定时限内的核酸检测阴性证明。

具体来时，如果在t时刻做了核酸检测，则经过一段时间后可以得到核酸检测阴性证明。这里我们假定等待核酸检测结果需要k个单位时间，即在t+k时刻可以获得结果。如果一个场所要求持24个单位时间内核酸检测结果入内，那么凭上述的核酸检测结果，可以在第t+k时刻到第t+k+23时刻进入该场所。

小C按时间顺序列出接下来的n项出行计划，其中第i项（1≤i≤n）可以概括为：

ti时刻进入某场所，该场所需持有ci个单位时间内的核酸检测结果入内，其中0＜ci≤2×105。

为了合理安排核酸检测时间，试根据小C的出行计划，回答如下查询：

如果在q时刻做了核酸检测，有多少项出行计划的核酸检测要求可以得到满足？

这样的查询共有m个，分别为q1,q2,…,qm；查询之间互不影响。

输入格式：

输入的第一行包含空格分隔的三个正整数n、m和k，分别表示出行计划数目、查询个数以及等待核酸检测结果所需时间。

接下来输入n行，其中每行包含用空格分隔的两个正整数ti、ci，表示一项出行计划；出行计划按时间顺序给出，满足0＜t1≤t2≤…≤tn≤2×105。

最后输入m行，每行仅包含一个正整数qi，表示一个查询。m个查询亦按照时间顺序给出，满足0＜q1＜q2＜…＜qm≤2×105。

输出格式：

输出共m行，每行一个整数，表示对应查询的答案。

样例输入：

6 2 10

5 24

10 24

11 24

34 24

35 24

35 48

1

2

样例输出：

3

3

5. CSP题目 碰撞的小球

问题描述：

数轴上有一条长度为L（L为偶数)的线段，左端点在原点，右端点在坐标L处。有n个不计体积的小球在线段上，开始时所有的小球都处在偶数坐标上，速度方向向右，速度大小为1单位长度每秒。

当小球到达线段的端点（左端点或右端点）的时候，会立即向相反的方向移动，速度大小仍然为原来大小。

当两个小球撞到一起的时候，两个小球会分别向与自己原来移动的方向相反的方向，以原来的速度大小继续移动。

现在，告诉你线段的长度L，小球数量n，以及n个小球的初始位置，请你计算t秒之后，各个小球的位置。

提示：

因为所有小球的初始位置都为偶数，而且线段的长度为偶数，可以证明，不会有三个小球同时相撞，小球到达线段端点以及小球之间的碰撞时刻均为整数。

同时也可以证明两个小球发生碰撞的位置一定是整数（但不一定是偶数）。

输入格式：

输入的第一行包含三个整数n, L, t，用空格分隔，分别表示小球的个数、线段长度和你需要计算t秒之后小球的位置。

第二行包含n个整数a1, a2, …, an，用空格分隔，表示初始时刻n个小球的位置。

输出格式：

输出一行包含n个整数，用空格分隔，第i个整数代表初始时刻位于ai的小球，在t秒之后的位置。

样例输入：

3 10 5

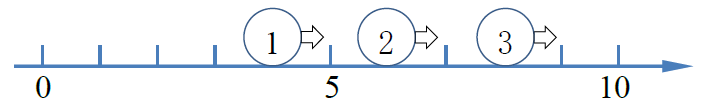
4 6 8

样例输出：

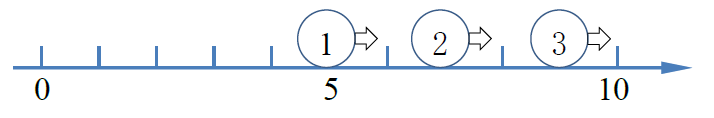
7 9 9

样例说明：

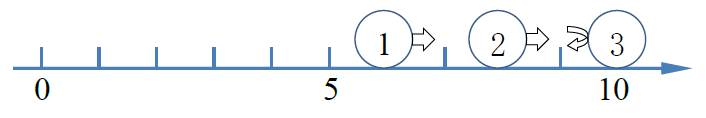
初始时，三个小球的位置分别为4, 6, 8。



一秒后，三个小球的位置分别为5, 7, 9。



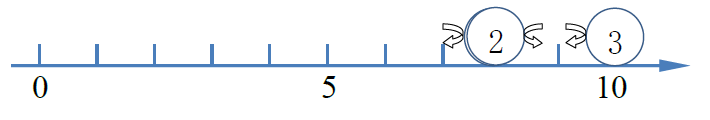
两秒后，第三个小球碰到墙壁，速度反向，三个小球位置分别为6, 8, 10。



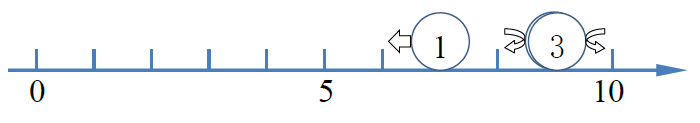
三秒后，第二个小球与第三个小球在位置9发生碰撞，速度反向（注意碰撞位置不一定为偶数），三个小球位置分别为7, 9, 9。



四秒后，第一个小球与第二个小球在位置8发生碰撞，速度反向，第三个小球碰到墙壁，速度反向，三个小球位置分别为8, 8, 10。



五秒后，三个小球的位置分别为7, 9, 9。



样例输入：

10 22 30

14 12 16 6 10 2 8 20 18 4

样例输出：

6 6 8 2 4 0 4 12 10 2

问题分析：该问题中，只需维护好每个小球的当前位置，并随时间进行变化，最后按题设要求进行输出即可，本题目实质是一道简单的模拟题，与树、图无关。

附加题：习题集 1.19 1.20 2.19

注：习题集为严蔚敏数据结构题集

**《数据结构》第2次上机题 （线性表练习）**

1. 设元素值为整型的线性表L，分别采用顺序结构和链式结构存储，编写函数，用选择/冒泡排序算法实现线性表的表排序。

2．设线性表A、B，元素值为整型，且递减有序，编写函数，实现下列功能：对采用顺序结构和链式结构2种存储结构，要求在A的空间上构成一个新线性表C，其元素为A和B元素的并集，且表C中的元素值递减有序（互不相同）。

3．输入正整数n、m（m<n），设有n个人坐成一圈，从第1个人开始循环报数，报到m的人出列，然后再从下一个人开始报数，报到m的人又出列，如此重复，直到所有的人都出列为止。要求用链式结构和顺序结构实现，按出列的先后顺序输出每个人的信息。

4．CSP题目

题目描述：  
新学期伊始，适逢顿顿书城有购书满x元包邮的活动，小P同学欣然前往准备买些参考书。

一番浏览后，小P初步筛选出n本书加入购物车中，其中第i本（1≤i≤n）的价格为ai元。

考虑到预算有限，在最终付款前小P决定再从购物车中删去几本书（也可以不删），使得剩余图书的价格总和m在满足包邮条件（m≥x）的前提下最小。

试帮助小P计算，最终选购哪些书可以在凑够x元包邮的前提下花费最小？

输入格式：  
从标准输入读入数据。

输入的第一行包含空格分隔的两个正整数n和x，分别表示购物车中图书数量和包邮条件。

接下来输入n行，其中第i行（1≤i≤n）仅包含一个正整数ai，表示购物车中第i本书的价格。输入数据保证n本书的价格总和不小于x。

输出格式：  
输出到标准输出。

仅输出一个正整数，表示在满足包邮条件下的最小花费。

样例1输入：

4 100

20

90

60

60

样例1输出：  
110  
样例1解释：  
购买前两本书（20+90）即可包邮且花费最小。

样例2输入：

3 30

15

40

30

样例2输出：  
30  
样例2解释：  
仅购买第三本书恰好可以满足包邮条件。

样例3输入：

2 90

50

50

样例3输出：  
100  
样例3解释：  
必须全部购买才能包邮。

5．CSP题目

问题描述：小H和小W来到了一条街上，两人分开买菜，他们买菜的过程可以描述为，去店里买一些菜然后去旁边的一个广场把菜装上车，两人都要买n种菜，所以也都要装n次车。具体的，对于小H来说有n个不相交的时间段[a1,b1],[a2,b2]…[an,bn]在装车，对于小W来说有n个不相交的时间段[c1,d1],[c2,d2]…[cn,dn]在装车。其中，一个时间段[s, t]表示的是从时刻s到时刻t这段时间，时长为t-s。  
由于他们是好朋友，他们都在广场上装车的时候会聊天，他们想知道他们可以聊多长时间。  
输入格式：  
输入的第一行包含一个正整数n，表示时间段的数量。  
接下来n行每行两个数ai，bi，描述小H的各个装车的时间段。  
接下来n行每行两个数ci，di，描述小W的各个装车的时间段。  
输出格式：  
输出一行，一个正整数，表示两人可以聊多长时间。  
样例输入：  
4  
1 3  
5 6  
9 13  
14 15  
2 4  
5 7  
10 11  
13 14  
样例输出：  
3  
数据规模和约定：  
对于所有的评测用例，1 ≤ n ≤ 2000, ai < bi < ai+1，ci < di < ci+1,对于所有的i(1 ≤ i ≤ n)有，1 ≤ ai, bi, ci, di ≤ 1000000。  
给两个人设定两个数组t[i]，t1[i] 当装车时置1，当t[i]=t1[i]时总数sum++。

附加题：习题集 2.24 2.29 2.30

**《数据结构》 第3次上机题（线性表复习，栈与队列练习）**

1．编程实现书P32 ADT Stack 基本操作9个，用顺序存储结构实现；

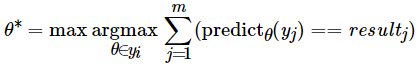
2．编程实现书P48 ADT Queue 基本操作9个，用链式存储结构实现；

3. 利用栈操作实现八皇后问题求解。

**4.** CSP题目  
题目描述：  
顿顿评估了m位同学上学期的安全指数，其中第i（1≤i≤m）位同学的安全指数为yi，是一个[0,108]范围内的整数；同时，该同学上学期的挂科情况记作resulti∈0,1，其中0表示挂科、1表示未挂科。  
相应地，顿顿用predictθ(y)表示根据阈值θ将安全指数y转化为的具体预测结果。  
如果predictθ(yj)与resultj相同，则说明阈值为θ时顿顿对第j位同学是否挂科预测正确；不同则说明预测错误。



最后，顿顿设计了如下公式来计算最佳阈值θ∗：



该公式亦可等价地表述为如下规则：

1. 最佳阈值仅在yi中选取，即与某位同学的安全指数相同；
2. 按照该阈值对这m位同学上学期的挂科情况进行预测，预测正确的次数最多（即准确率最高）；
3. 多个阈值均可以达到最高准确率时，选取其中最大的。

输入格式：  
从标准输入读入数据。  
输入的第一行包含一个正整数m。  
接下来输入m行，其中第i（1≤i≤m）行包括用空格分隔的两个整数yi和resulti，含义如上文所述。

输出格式：  
输出到标准输出。  
输出一个整数，表示最佳阈值θ∗。

样例1输入：  
6  
0 0  
1 0  
1 1  
3 1  
5 1  
7 1  
样例1输出：  
3  
样例1解释：  
按照规则一，最佳阈值的选取范围为0,1,3,5,7。  
θ=0时，预测正确次数为4；  
θ=1时，预测正确次数为5；  
θ=3时，预测正确次数为5；  
θ=5时，预测正确次数为4；  
θ=7时，预测正确次数为3。  
阈值选取为1或3时，预测准确率最高；  
所以按照规则二，最佳阈值的选取范围缩小为1,3。  
依规则三，θ∗=max1,3=3。

样例2输入：  
8  
5 1  
5 0  
5 0  
2 1  
3 0  
4 0  
100000000 1  
1 0  
样例2输出：  
100000000

子任务：  
70%的测试数据保证m≤200；  
全部的测试数据保证2≤m≤105。

5. CSP题目

问题描述：

某次测验后，顿顿老师在黑板上留下了一串数字23333便飘然而去。凝望着这个神秘数字，小P同学不禁陷入了沉思……

已知某次测验包含n道单项选择题，其中第i题（1≤i≤n）有ai个选项，正确选项为bi，满足ai≥2且0≤bi＜ai。比如说，ai=4表示第i题有4个选项，此时正确选项bi的取值一定是0、1、2、3其中之一。

顿顿老师设计了如下方式对正确答案进行编码，使得仅用一个整数m便可表示b1,b2,…,bn。

首先定义一个辅助数组ci，表示数组ai的前缀乘积。当1≤i≤n时，满足：

特别地，定义c0=1。

于是m便可按照如下公式算出：

易知，0≤m＜cn，最小值和最大值分别当bi全部为0和bi=ai-1时取得。

试帮助小P同学，把测验的正确答案b1,b2,…,bn从顿顿老师留下的神秘整数m中恢复出来。

输入格式：

从标准输入读入数据。

输入共两行。

第一行包含用空格分隔的两个整数n和m，分别表示题目数量和顿顿老师的神秘数字。

第二行包含用空格分隔的n个整数a1,a2,…,an，依次表示每道选择题的选项数目。

输出格式：

输出到标准输出。

输出仅一行，包含用空格分隔的n个整数b1,b2,…,bn，依次表示每道选择题的正确选项。

样例1输入：

15 32767

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

样例1输出：

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

样例2输入：

4 0

2 3 2 5

样例2输出：

0 0 0 0

样例3输入：

7 23333

3 5 20 10 4 3 10

样例3输出：

2 2 15 7 3 1 0

附加题：习题集 2.32 2.37 2.38

**《数据结构》 第4次上机题（线性结构练习）**

1．输入稀疏矩阵，建立稀疏矩阵三元组顺序结构，实现矩阵的列序遍历转置和快速转置算法。

2. 求矩阵的马鞍点。（书P69 7）

3. CSP 题目

问题描述：

一次放学的时候，小明已经规划好了自己回家的路线，并且能够预测经过各个路段的时间。同时，小明通过学校里安装的“智慧光明”终端，看到了**出发时刻路上**经过的所有红绿灯的指示状态。请帮忙计算小明此次回家所需要的时间。

输入格式：

输入的第一行包含空格分隔的三个正整数 r、y、g，表示红绿灯的设置。这三个数均不超过 106。  
输入的第二行包含一个正整数 n，表示小明总共经过的道路段数和路过的红绿灯数目。  
接下来的 n 行，每行包含空格分隔的两个整数 k、t。k=0 表示经过了一段道路，将会耗时 t 秒，此处 t 不超过 106；k=1、2、3 时，分别表示**出发时刻**，此处的红绿灯状态是红灯、黄灯、绿灯，且倒计时显示牌上显示的数字是 t，此处 t 分别不会超过 r、y、g。

输出格式：

输出一个数字，表示此次小明放学回家所用的时间。

样例输入：

30 3 30  
8  
0 10  
1 5  
0 11  
2 2  
0 6  
0 3  
3 10  
0 3

样例输出：

46

样例说明：

小明先经过第一段路，用时10秒。第一盏红绿灯出发时是红灯，还剩5秒；小明到达路口时，这个红绿灯已经变为绿灯，不用等待直接通过。接下来经过第二段路，用时11秒。第二盏红绿灯出发时是黄灯，还剩两秒；小明到达路口时，这个红绿灯已经变为红灯，还剩11秒。接下来经过第三、第四段路，用时9秒。第三盏红绿灯出发时是绿灯，还剩10秒；小明到达路口时，这个红绿灯已经变为红灯，还剩两秒。接下来经过最后一段路，用时3秒。共计10+11+11+9+2+3=46秒。

评测用例规模与约定：

　　有些测试点具有特殊的性质：  
　　\* 前2个测试点中不存在任何信号灯。  
　　测试点的输入数据规模：  
　　\* 前6个测试点保证n≤103。  
　　\* 所有测试点保证n≤105。

4. CSP题目

问题描述：

西西艾弗岛的购物中心里店铺林立，商品琳琅满目。为了帮助游客根据自己的预算快速选择心仪的商品，IT 部门决定研发一套商品检索系统，支持对任意给定的预算x，查询在该预算范围内（≤x）价格最高的商品。如果没有商品符合该预算要求，便向游客推荐可以免费领取的西西艾弗岛定制纪念品。

假设购物中心里有n件商品，价格从低到高依次为A1,A2,…,An，则根据预算x检索商品的过程可以抽象为如下序列查询问题。

A=[A0,A1,A2,…,An]是一个由n+1个[0,N)范围内整数组成的序列，满足0= A0＜A1＜A2＜…＜An＜N。（这个定义中蕴含了n一定小于N。）

基于序列A，对于[0,N)范围内任意的整数x，查询f(x)定义为：序列A中小于等于x的整数里最大的数的下标。具体来说有以下两种情况：

(1)存在下标0≤i＜n满足Ai≤x＜Ai+1

此时序列A中从A0到Ai均小于等于x，其中最大的数为Ai，其下标为i，故f(x)=i。

(2) An≤x

此时序列A中所有的数都小于等于x，其中最大的数为An，故f(x)=n。

令sum(A)表示f(0)到f(N-1)的总和，即：

对于给定的序列A，试计算sum(A)。

输入格式：

从标准输入读入数据。

输入的第一行包含空格分隔的两个正整数n和N。

输入的第二行包含n个用空格分隔的整数A1,A2,…,An。

注意A0固定为0，因此输入数据中不包括A0。

输出格式：

输出到标准输出。

仅输出一个整数，表示sum(A)的值。

样例1输入：

3 10  
2 5 8

样例1输出：

15

样例2输入：

9 10

1 2 3 4 5 6 7 8 9

样例2输出：

45

附加题：习题集 3.20 3.28 3.32

**第一阶段总结**

**第二阶段（树与图部分）**

**《数据结构》 第5次上机题目 (二叉树练习 )**

1. 编程实现书P75 ADT BinaryTree 基本操作20个，用二叉链表结构实现；

2.实现二叉树的先序、中序、后序遍历，用递归和非递归方法；实现层次遍历。

3. 设二叉树采用二叉链表存储，编写函数，对二叉树中每个元素值为x的结点，删除以它为根的子树，并释放相应空间。（习题集6.45）

4. 编写函数，判断给定的二叉树是否是完全二叉树。（习题集6.49）

5． CSP题目

问题描述：

一个未经初始化的变量，里面存储的值可能是任意的。因此直接使用未初始化的变量，比如将其赋值给另一个变量，并不符合一般的编程逻辑。代码中出现这种情况，往往是因为遗漏了初始化语句、或是打错了变量名。对代码中使用了未初始化变量的语句进行检查，可以方便地排查出代码中的一些隐秘 Bug。

考虑一段包含k条赋值语句的简单代码。该段代码最多使用到n个变量，分别记作a1,a2,…,an；该段代码使用的常量均记作a0。

第i条（1≤i≤k）赋值语句为axi=ayi，满足1≤xi≤n、0≤yi≤n，表示将ayi的值赋给变量axi。其中axi被称为该赋值语句的左值，一定是个变量；ayi被称为右值，可以是一个常量或变量。

对于任意一条赋值语句axi=ayi，如果右值ayi是一个变量，则其应该在此之前被初始化过。

具体来说，如果变量ayi在前i-1条赋值语句中做为左值出现过，即存在j＜i满足xj=yi（这里无需考虑第j条赋值语句本身是否也有右值未初始化的问题），我们就认为在第i条赋值语句中ayi已被初始化；

否则，我们认为该条语句存在右值未初始化的问题。

按照上述规则，试统计给定的代码中，有多少条赋值语句右值未被初始化。

输入格式：  
输入的第一行包含空格分隔的两个正整数n、k，分别表示变量的数量和赋值语句的条数。

接下来输入k行，其中第i行（1≤i≤k）包含空格分隔的两个正整数xi、yi，表示第i条赋值语句。

输出格式：

输出一个整数，表示有右值未被初始化问题的赋值语句条数。

样例输入：  
10 7

1 2

3 3

3 0

3 3

6 2

2 1

8 2

样例输出：

3

样例解释：

其中第一、二、五条赋值语句右值未被初始化。

附加题：习题集 6.42 6.45 6.47

**《数据结构》 第6次上机题目 ( 二叉树、树、图练习 )**

1．编程实现书P96 ADT Graph 基本操作11个，用邻接矩阵存储结构实现；

2. 输入N个权值（1-100正整数），建立哈夫曼树。

3. 编写函数，对二叉链表结构的二叉树，求宽度。（书P94 4）

4. 编写函数，对一棵以孩子-兄弟链表表示的树，输出第i层的所有元素。

5. CSP题目

问题描述：

俄罗斯方块是俄罗斯人阿列克谢·帕基特诺夫发明的一款休闲游戏。

游戏在一个15行10列的方格图上进行，方格图上的每一个格子可能已经放置了方块，或者没有放置方块。每一轮，都会有一个新的由4个小方块组成的板块从方格图的上方落下，玩家可以操作板块左右移动放到合适的位置，当板块中某一个方块的下边缘与方格图上的方块上边缘重合或者达到下边界时，板块不再移动，如果此时方格图的某一行全放满了方块，则该行被消除并得分。

在这个问题中，你需要写一个程序来模拟板块下落，你不需要处理玩家的操作，也不需要处理消行和得分。

具体的，给定一个初始的方格图，以及一个板块的形状和它下落的初始位置，你要给出最终的方格图。

输入格式：

输入的前15行包含初始的方格图，每行包含10个数字，相邻的数字用空格分隔。如果一个数字是0，表示对应的方格中没有方块，如果数字是1，则表示初始的时候有方块。输入保证前4行中的数字都是0。

输入的第16至第19行包含新加入的板块的形状，每行包含4个数字，组成了板块图案，同样0表示没方块，1表示有方块。输入保证板块的图案中正好包含4个方块，且4个方块是连在一起的（准确的说，4个方块是四连通的，即给定的板块是俄罗斯方块的标准板块）。

第20行包含一个1到7之间的整数，表示板块图案最左边开始的时候是在方格图的哪一列中。注意，这里的板块图案指的是16至19行所输入的板块图案，如果板块图案的最左边一列全是0，则它的左边和实际所表示的板块的左边是不一致的（见样例）

输出格式：

输出15行，每行10个数字，相邻的数字之间用一个空格分隔，表示板块下落后的方格图。注意，你不需要处理最终的消行。

样例输入：

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 1 0 0 0

1 1 1 0 0 0 1 1 1 1

0 0 0 0 1 0 0 0 0 0

0 0 0 0

0 1 1 1

0 0 0 1

0 0 0 0

3

样例输出：

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 1 0 0 0

1 1 1 1 1 1 1 1 1 1

0 0 0 0 1 1 0 0 0 0

附加题：习题集 6.49 6.60 6.62

**《数据结构》 第7次上机题目 ( 图 练习 )**

1. 图的深度优先和广度优先遍历；

2．编程实现Dijkstra算法；

3．CSP题目

题目描述：

小明在他的果园里种了一些苹果树，这些苹果树排列成一个圆。为了保证苹果的品质，在种植过程中要进行疏果操作。为了更及时地完成疏果操作，小明会不时地检查每棵树的状态，根据需要进行疏果。检查时，如果发现可能有苹果从树上掉落，小明会重新统计树上的苹果个数（然后根据之前的记录就可以判断是否有苹果掉落了)。在全部操作结束后，请帮助小明统计相关的信息。

输入格式：

从标准输入读入数据。

第1行包含一个正整数N，表示苹果树的棵数。

第1+i行（1≤i≤N)，每行的格式为mi,ai1,ai2,... ,aimi,。其中，第一个正整数mi表示本行后面的整数个数。后续的mi个整数表示小明对第i棵苹果树的操作记录。若aij (1≤ j ≤mi）为正整数，则表示小明进行了重新统计该棵树上的苹果个数的操作，统计的苹果个数为aij;若为零或负整数，则表示一次疏果操作，去掉的苹果个数是|aij|。

输入保证一定是正确的，满足:

1. ai1>0，即对于每棵树的记录，第一个操作一定是统计苹果个数（初始状态，此时不用判断是否有苹果掉落）；
2. 每次疏果操作保证操作后树上的苹果个数仍为正。

输出格式：

输出到标准输出。

输出只有一行，包含三个整数T、D、E。其中，

1. T为全部疏果操作结束后所有苹果树上剩下的苹果总数（假设每棵苹果树在最后一次统计苹果个数操作后苹果不会因为疏果以外的原因减少);
2. D为发生苹果掉落的苹果树的棵数;
3. E为相邻连续三棵树发生苹果掉落情况的组数。

对于第三个统计量的解释：N棵苹果树A1,A2,...,AN排列成一个圆，那么A1与A2相邻，A2与A3相邻，......，AN-1与AN相邻，AN与A1相邻。如果Ai-1, Ai，Ai+1这三棵树都发生了苹果掉落的情况，则记为一组。形式化的，有



其中，Drop(Ai）表示苹果树Ai是否发生苹果掉落的情况，Pred(Ai)表示Ai的前一棵树Ai-1（如果i>1）或者AN（如果i=1)，Succ(Ai)表示Ai的后一棵树Ai+1（如果i<N）或者A1（如果i=N）

样例1输入：

4

4 74 -7 -12 -5

5 73 -8 -6 59 -4

5 76 -5 -10 60 -2

5 80 -6 -15 59 0

样例1输出：

222 1 0

样例1解释：

全部操作结束后，第1棵树上剩下的苹果个数为74-7-12-5=50，第2棵为59-4=55，第3棵为60-2=58，第4棵为59-0=59。因此T=50+55+58+59=222。

其中，第3棵树在第2次统计之前剩下的苹果个数为76-5-10=61>60，因此发生了苹果掉落的情况。可以检验其他的树没有这种情况，因此D=1。

没有连续三棵树都发生苹果掉落的情况，因此E=0。

样例2输入：

5

4 10 0 9 0

4 10 -2 7 0

2 10 0

4 10 -3 5 0

4 10 -1 8 0

样例2输出：

39 4 2

样例2解释：

第1、2、4、5棵树发生了苹果掉落的情况，因此D=4。其中，连续三棵树都发生苹果掉落情况的有(5,1,2)和(4,5,1)，因此E=2。

4．CSP题目

问题描述：

暑假要到了。可惜由于种种原因，小P原本的出游计划取消。失望的小P只能留在西西艾弗岛上度过一个略显单调的假期……直到……

某天，小P获得了一张神秘的藏宝图。

西西艾弗岛上种有n棵树，这些树的具体位置记录在一张绿化图上。

简单地说，西西艾弗岛绿化图可以视作一个大小为(L+1)×(L+1)的01矩阵A，

地图左下角（坐标(0,0)）和右上角（坐标(L,L)）分别对应A[0][0]和A[L][L]。

其中A[i][j]=1表示坐标(i,j)处种有一棵树，A[i][j]=0则表示坐标(i,j)处没有树。

换言之，矩阵A中有且仅有的n个1展示了西西艾弗岛上n棵树的具体位置。

传说，大冒险家顿顿的宝藏就埋藏在某棵树下。

并且，顿顿还从西西艾弗岛的绿化图上剪下了一小块，制作成藏宝图指示其位置。

具体来说，藏宝图可以看作一个大小为(S+1)×(S+1)的01矩阵B（S远小于L），对应着A中的某一部分。

理论上，绿化图A中存在着一处坐标(x,y)（0≤x,y≤L-S）与藏宝图B左下角(0,0)相对应，即满足：对B上任意一处坐标(i,j)（0≤i,j≤S），都有A[x+i][y+j]=B[i][j]。

当上述条件满足时，我们就认为藏宝图B对应着绿化图A中左下角为(x,y)、右上角为(x+S,y+S)的区域。

实际上，考虑到藏宝图仅描绘了很小的一个范围，满足上述条件的坐标(x,y)很可能存在多个。

请结合西西艾弗岛绿化图中n棵树的位置，以及小P手中的藏宝图，判断绿化图中有多少处坐标满足条件。

特别地，藏宝图左下角位置一定是一棵树，即A[x][y]=B[0][0]=1，表示了宝藏埋藏的位置。

输入格式：

从标准输入读入数据。

输入的第一行包含空格分隔的三个正整数n、L和S，分别表示西西艾弗岛上树的棵数、绿化图和藏宝图的大小。

由于绿化图尺寸过大，输入数据中仅包含n棵树的坐标而非完整的地图；即接下来n行每行包含空格分隔的两个整数x和y，表示一棵树的坐标，满足0≤x,y≤L且同一坐标不会重复出现。

最后(S+1)行输入小P手中完整的藏宝图，其中第i行（0≤i≤S）包含空格分隔的(S+1)个0和1，表示B[S-i][0]…B[S-i][S]。

需要注意，最先输入的是B[S][0]…B[S][S]一行，B[0][0]…B[0][S]一行最后输入。

输出格式：

输出到标准输出。

输出一个整数，表示绿化图中有多少处坐标可以与藏宝图左下角对应，即可能埋藏着顿顿的宝藏。

样例1输入:

5 100 2

0 0

1 1

2 2

3 3

4 4

0 0 1

0 1 0

1 0 0

样例1输出:

3

样例1解释：

绿化图上(0,0)、(1,1)和(2,2)三处均可能埋有宝藏。

样例2输入:

5 4 2

0 0

1 1

2 2

3 3

4 4

0 0 0

0 1 0

1 0 0

样例2输出:

0

样例2解释：

如果将藏宝图左下角与绿化图(3,3)处对应，则藏宝图右上角会超出绿化图边界，对应不成功。

附加题：习题集 7.27 7.34 7.37

**第二阶段总结**

**第三阶段（查找与排序部分）**

**《数据结构》 第8次上机题目 ( 查找 排序 练习 )**

1. 实现二叉排序树的插入和删除。

2．实现交换、选择、归并等简单排序算法；

3．实现快速排序算法；

4．实现堆排序算法；

5．CSP题目

题目背景：

开学了，可是校园里堆积了不少垃圾杂物。

热心的同学们纷纷自发前来清理，为学校注入正能量～

题目描述：

通过无人机航拍我们已经知晓了n处尚待清理的垃圾位置，其中第i (1≤i≤n)处的坐标为(xi,yi)，保证所有的坐标均为整数。

我们希望在垃圾集中的地方建立些回收站。具体来说，对于一个位置(x,y)是否适合建立回收站，我们主要考虑以下几点：

1. (x,y)必须是整数坐标，且该处存在垃圾；
2. 上下左右四个邻居位置，即(x,y+1)、(x,y-1)、(x+1,y)和(x-1,y)处，必须全部存在垃圾；
3. 进一步地，我们会对满足上述两个条件的选址进行评分分数为不大于4的自然数，表示在(x±1,y±1)四个对角位置中有几处存在垃圾。

现在，请你统计一下每种得分的选址个数。

输入格式：

从标准输入读入数据。

输入总共有n+1行。

第1行包含一个正整数n，表示已查明的垃圾点个数。

第1+i行（1≤i≤n）包含由一个空格分隔的两个整数xi和yi，表示第i处垃圾的坐标。

保证输入的n个坐标互不相同。

输出格式：

输出到标准输出。

输出共五行，每行一个整数，依次表示得分为0、1、2、3和4的回收站选址个数。

样例1输入：

7

1 2

2 1

0 0

1 1

1 0

2 0

0 1

样例1输出：

0

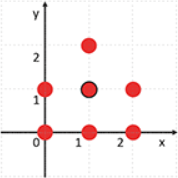
0

1

0

0

样例1解释：



如图所示，仅有(1,1)可选为回收站地址，评分为2。

样例2输入：

2

0 0

-100000 10

样例2输出：

0

0

0

0

0

样例2解释：

不存在可选地址。

样例3输入：

11

9 10

10 10

11 10

12 10

13 10

11 9

11 8

12 9

10 9

10 11

12 11

样例3输出：

0

2

1

0

0

样例3解释：

1分选址：(10,10)和(12,10);

2分选址：(11,9)。

提示：

本题中所涉及的坐标皆为整数，且保证输入的坐标两两不同。

附加题：习题集 9.32 10.32 10.34

**第三阶段总结**