**“王选杯”信息学竞赛**

**时间：2015年5月30日13：00—15：30**

**建立一个文件夹：学校+姓名，例如：八十中xxx**

**本次测试是文件读写, 样例：**

**Freopen(“toy.in”,”r”,stdin);**

**Freopen(“toy.out”,”w”,stdout);**

A. 玩具车

时间限制：1秒

内存限制：256MB

输入文件：toy.in

输出文件：toy.out

【问题描述】

阿米巴有N种玩具车，每种玩具车有足够多个，他想测试一下哪几种车是好的。一种玩具车是好的，当且仅当它在所有的碰撞中都没有损坏。每两种玩具车都会碰撞一次。碰撞的结果在一个的矩阵中表示出来，矩阵的第i行第j列的元素Aij表示：

-1：碰撞没有发生。-1只在主对角线上出现。

0：没有车在碰撞中损坏。

1：只有第i种玩具车损坏。

2：只有第j种玩具车损坏。

3：两个玩具车都在碰撞中损坏。

请你写出有哪几种玩具车是好的。

【输入数据】

第一行是一个整数N（1<=N<=100），玩具车的种类。

接下来有N行，每行N个整数，表示矩阵A，矩阵A的意义如题目表述所示。

输入数据保证-1只出现在矩阵的主对角线上。

输入数据保证合法，即如果Aij=0则Aji=0；如果Aij=1则Aji=2；如果Aij=2则Aji=1；如果Aij=3则Aji=3。

【输出数据】

第一行为好的玩具车的种类数。

如果好的玩具车的种类数不为0，在第二行写出好的玩具车的种类标号，按升序排列，以空格分隔。

【样例输入1】

3

-1 0 0

0 -1 1

0 2 -1

【样例输出1】

2

1 3

【样例输入2】

4

-1 3 3 3

3 -1 3 3

3 3 -1 3

3 3 3 -1

【样例输出2】

0

B. 密码锁

时间限制：1秒

内存限制：256MB

输入文件：lock.in

输出文件：lock.out

【问题描述】

阿米巴有一把密码锁，由N个可以旋转的部分组成。每个部分的数字都是0到9，如图所示。每一步，阿米巴可以顺时针旋转一个部分或者逆时针旋转一个部分。如果要打开一个密码锁，只要将每一部分的数字与密码相一致即可。现在告诉你密码锁的当前状态和密码，问最少需要多少步才能打开这把锁？



【输入数据】

第一行为一个整数N（1<=N<=1000），表示数字锁的部分数。

第二行为一个有N个数字的字符串，表示密码锁的当前状态。

第三行为一个有N个数字的字符串，表示密码锁的密码。

【输出数据】

一行，一个整数，表示最少步数。

【样例输入】

5

82195

64723

【样例输出】

13

【样例解释】

样例总共需要13步。

第一部分：8->7->6

第二部分：2->3->4

第三部分：1->0->9->8->7

第四部分：9->0->1->2

第五部分：5->4->3

C. 等价字符串

时间限制：1秒

内存限制：256MB

输入文件：string.in

输出文件：string.out

【问题描述】

阿米巴手里有两个01串。两个等长的01串的编辑距离被定义为，在同一位上有不同数字的位数。等价字符串是指一个等长的01串，其与两个目标01串的编辑距离相等。阿米巴很感兴趣他手中两个字符串的等价字符串，你能帮他找出一个吗？

【输入数据】

第一行为一个长度为N的01串，表示阿米巴手里的第一个01串。

第二行为一个长度为N的01串，表示阿米巴手里的第二个01串。

1<=N<=100000

【输出数据】

输出只有一行，如果存在满足要求的答案，输出一行，为一个长度为N的01串。如果没有满足要求的答案，输出“Impossible”（不包含引号）。如果有多组解，输出任意一个。

【样例输入1】

0001

1011

【样例输出1】

0011

【样例输入2】

000

111

【样例输出2】

Impossible

【样例解释】

在第一个样例中，共有以下01串都是满足题目要求的：0010, 0011, 0110, 0111, 1000, 1001, 1100, 1101。

D. 考试成绩

时间限制：1秒

内存限制：256MB

输入文件：score.in

输出文件：score.out

【问题描述】

阿米巴在需要一共要进行N场考试，每场考试的成绩可以是1到P的任意整数。阿米巴非常聪明，所以他可以考出任意一个他希望得到的分数。阿米巴不想他的成绩太好，如果他考试分数的总和大于X，那么他的同学会嫉妒他；阿米巴也不想他的成绩太差，如果他考试分数的中位数小于Y，阿米巴的妈妈会认为阿米巴不好好学习。现在阿米巴已经进行了K场考试，分数分别是a1, a2, …, ak。现在阿米巴希望既不被他的同学嫉妒，也不希望被他的妈妈批评，那么他在之后的N-K场考试应该考多少分数？

【输入数据】

第一行包含五个整数N, K, P, X, Y (1<=N<=999，N是奇数，0<=K<N，1<=P<=1000，N<=X<=N\*P，1<=Y<=P)。N, K, P, X, Y的意义如题目描述中所示。

第二行包含k个整数，用空格分开，分别代表a1, a2, …, ak (1<=ai<=p)。

【输出数据】

如果阿米巴无法达到两个要求，输出“-1”（不包含引号）。否则输出N-K个用空格分隔的整数，代表剩余考试的考试分数。如果有多组解，可以输出任意一组。

【样例输入1】

5 3 5 18 4

3 5 4

【样例输出1】

4 1

【样例输入2】

5 3 5 16 4

5 5 5

【样例输出2】

-1

【样例解释】

在第一组样例中，样例输出的结果是3+5+4+4+1=17，不超过18；1, 3, 4, 4, 5的中位数是4，不小于4，所以满足两个要求。"4 2", "2 4", "5 1", "1 5", "4 1", "1 4" 这些也都是样例1的答案。

在第二组样例中，阿米巴已经得到了3个5分，即使之后获得两个1分，总分也是17分，超过了16分，所以输出是-1。

E. 砍树

时间限制：1秒

内存限制：256MB

输入文件：wood.in

输出文件：wood.out

【问题描述】

阿米巴是森林里一个快乐的伐木工。森林里有N棵树，坐落在坐标轴上，任意两棵树的位置都不相同。每棵树有一个坐标Xi，有一个高度Hi。一棵树被砍倒可以向左倒或者向右倒，向左倒会占据[Xi-Hi, Xi]这个区间，向右倒会占据[Xi, Xi+Hi]这个区间。一个树可以被砍倒，当且仅当在它倒下的占据的区间里没有任意一个点被占据。每棵树在没被砍倒的情况下也会占据Xi这个点。阿米巴想知道，他最多能砍倒多少棵树。

【输入数据】

第一行包含一个整数N (1<=N<=100000)，树的个数。

接下来N行，每行两个整数Xi, H­i，代表树的位置和高度。

输入数据保证Xi两两不同，并且按递增顺序排列。

【输出数据】

输出一个整数，在给定规则下能够砍下的最多的树的个数。

【样例输入1】

5

1 2

2 1

5 10

10 9

19 1

【样例输出1】

3

【样例输入2】

5

1 2

2 1

5 10

10 9

20 1

【样例输出2】

4

【样例注释】

第一个样例的一组实现方式是：

砍第一颗树，占据[-1, 1]；

砍第二颗树，占据[2, 3]；

不砍第三棵树，占据点5；

不砍第四颗树，占据点10；

砍第五棵树，占据[19, 20]。

F. 洞穴

时间限制：1秒

内存限制：256MB

输入文件：cave.in

输出文件：cave.out

【问题描述】

阿米巴在玩一个电脑游戏，每一关都会身处于一个洞穴之中。这个洞穴可以由一个N\*M个格子组成，自上向下共有N行，自左向右共有M列，共有N\*M个格子，把第r行第c列的格子称为(r, c)。

每个格子有两种状态，破碎的或者有薄冰的。如果经过一个破碎的格子，他将会掉下去；如果经过一个有薄冰的格子，这个格子会变成破碎的。阿米巴最初的位置是(r1, c1)，由于他是从上一关掉下来的，所以保证(r1, c1)是一个破碎的格子，但阿米巴不会因初始格子是一个破碎的格子而直接掉下去；若阿米巴需要从(r1, c1)掉下去，需要再次访问(r1, c1)。如果他从(r2, c2)的地方掉到下去，他将顺利通过这一关。阿米巴试了很久都没有成功，现在他想知道，当前的关卡是否是有解的呢？

【输入数据】

第一行有两个整数N，M（1<=N<=500, 1<=M<=500），代表洞穴的行数和列数。

接下来有N行，每行有M个字符，表示每个格子的状态。每个字符可能是“.”，表示这个格子是有薄冰的；或者是“X”，表示这个格子是破碎的。

接下来的一行有两个整数，r1，c1，表示初始地点。输入数据保证(r1, c1)是一个破碎的格子。

接下来的一行有两个整数，r2，c2，表示需要掉下去的地点。掉下去的地点可能和初始地点相同。

【输出数据】

如果能完成目标，输出“YES”（不含引号），否则输出“NO”（不含引号）。

【样例输入1】

4 6

X...XX

...XX.

.X..X.

......

1 6

2 2

【样例输出1】

YES

【样例输入2】

5 4

.X..

...X

X.X.

....

.XX.

5 3

1 1

【样例输出2】

NO

【样例输入3】

4 7

..X.XX.

.XX..X.

X...X..

X......

2 2

1 6

【样例输出3】

YES

G. 队列

时间限制：1秒

内存限制：256MB

输入文件：queue.in

输出文件：queue.out

【问题描述】

阿米巴现在在玩一个模拟经营类游戏。现在一共有N个顾客，第i个顾客需要的服务时间是Ti。一个顾客会感到失望，当且仅当他等待的时间大于他所需要的服务时间。所有的顾客会被安排在一个队列里被服务，一个顾客的等待时间是在他之前的顾客的所有服务时间之和。阿米巴想要知道，在最优情况下最多会有多少个顾客不失望？

【输入数据】

第一行是一个整数N（1<=N<=100000），表示顾客的数量。

第二行包含N个整数ti（1<=ti<=109），表示每个顾客服务所需的时间。

【输出数据】

一行，表示不失望的顾客的最多的数量。

【样例输入】

5

15 2 1 5 3

【样例输出】

4

【样例解释】

可以将队列安排为1, 2, 3, 5, 15，除了服务时间为5的顾客以外都可以不失望。

H. 比赛

时间限制：1秒

内存限制：256MB

输入文件：competition.in

输出文件：competition.out

【问题描述】

阿米巴的班级里进行一场奇怪的石头剪子布的比赛。阿米巴的班级里总共有3类人，有R个人只出石头，有S个人只出剪刀，有P个人只出布。最初所有的人都存活。每一时刻，会从仍然存活的人中等概率的选出两个人，出石头的人胜出剪刀的人，出剪刀的人胜出布的人，出布的人胜出石头的人，输的人就将被淘汰，赢或平的人将继续存活。在足够长的时间以后会只剩下一个人，阿米巴关心这个人是出石头的、出剪刀的还是出布的呢？

【输入数据】

一行，有三个整数，R，S，P(1<=R<=100, 1<=S<=100, 1<=P<=100)，代表最初的三种人的个数。

【输出数据】

一行，有三个实数，分别代表剩下的人出石头、出剪刀、出布的人的概率。答案被认为是正确的当且仅当相对误差或绝对误差小于10-9。

【样例输入1】

2 2 2

【样例输出1】

0.333333333333 0.333333333333 0.333333333333

【样例输入2】

2 1 2

【样例输出2】

0.150000000000 0.300000000000 0.550000000000

【样例输入3】

1 1 3

【样例输出3】

0.057142857143 0.657142857143 0.285714285714

I. 最短路树

时间限制：3秒

内存限制：256MB

输入文件：path.in

输出文件：path.out

【问题描述】

阿米巴也是信息学竞赛的参与者。这天他在学习单源最短路，他发现从起点出发的最短路可以形成一颗树的形式。形式化地说，对于一个联通带权无向图G=(V, E)，它的最短路树是图G的一个生成树G1=(V, E1)，并且从起点u出发到任意一个顶点的最短路的距离在G和G1上是相等的。阿米巴现在想知道，在最短路树中边权总和最小的最短路树是哪一棵？

【输入数据】

第一行包含两个整数N, M（1<=N<=300000, 0<=M<=300000），分别代表图G的点数和边数。

接下来有M行，每行三个整数ui, vi, wi（1<=ui, vi<=N, 1<=wi<=109, ui!=vi），分别代表一条边的起点、终点和边权。

最后一行包含一个整数u，表示最短路的起点。

输入数据保证任意两对顶点间最多有一条边。

【输出数据】

第一行为一个整数，为最小的最短路树的边权和。

第二行为共有N-1个整数，位这个最短路树的边的编号，用空格分隔。边的编号是由根据输入数据顺序从1开始标号。边的标号的输出顺序是任意的。如果存在多解，可以任意输出一个。

【样例输入1】

3 3

1 2 1

2 3 1

1 3 2

3

【样例输出1】

2

1 2

【样例输入2】

4 4

1 2 1

2 3 1

3 4 1

4 1 2

4

【样例输出2】

4

2 3 4

J. 逆序对

时间限制：2秒

内存限制：256MB

输入文件：inversion.in

输出文件：inversion.out

【问题描述】

阿米巴学会了用归并排序求逆序对数，但是他并不满足于这个算法，他现在考虑这样一个问题：初始有一个无穷序列p={1, 2, 3, …}，然后进行了N次交换操作(ai, bi)，代表交换序列中的第ai个元素和第bi个元素。阿米巴想知道，在经过了N次操作以后的序列的逆序对的个数是多少。逆序对的个数是满足i>j和pi<p­j的有序数对(i, j)的个数。

【输入数据】

第一行包含一个整数N(1<=N<=100000)，表示交换操作的个数。

接下来有N行，每行包含两个整数，ai, bi，表示第i次交换操作交换的位置。

【输出数据】

一行，表示逆序对的个数。

【样例输入1】

2

4 2

1 4

【样例输出1】

4

【样例输入2】

3

1 6

3 4

2 5

【样例输出2】

15

【样例解释】

在第一个样例中，初始的序列是{1, 2, 3, 4, 5, …}，经过第一个交换操作是{1, 4, 3, 2, 5, …}，经过第二个交换操作是{2, 4, 3, 1, 5, …}，这个序列共有4个逆序对(1, 4), (2, 3), (2, 4), (3, 4)。