**DP习题**

**Poj1018**

我们从 Pizoor Communication Inc 得到一个订单。这个序列是一个特殊的通信系统，该系统包含了多个设备，对于每一个设备我们都可以可从几个不同的厂商那里获得。而每个厂商的同种设备主要存在两个方面的差别：bandwidths和prices.我们定义所有的bandwidths中最小的为B，prices和为P。我们的目标是求出使得B/P的值最大的方案。

**输入**

第一行输入一个整数t(1<=t<=10)表示测试数据的组数。第二行n(1<=n<=100)表示需要不同类型设备的总数量。接下来n行第一个数字表示该公司拥有的设备总数，接着就是每一种设备的bandwidth及对应的price。

**输出**

输出B/P的最大值，保留3位小数。

**样例输入**

1

3

3 100 25 150 35 80 25

2 120 80 155 40

2 100 100 120 110

**样例输出**

0.649

**Poj 1651**

乘法谜题是用一排卡片，每一个包含一个正整数。在移动过程中，玩家取出一张卡片，并用所取数字的点数与它左边卡片数字和右边卡片的数字相乘。在该行中，不允许取出第一个和最后一个卡。最后一步行动后，只有两个卡被留在。

目标是以这样的顺序，以尽量减少得分点的总数。

例如，如果在该行中的卡包含数字10 1 50 20 5，玩家可能需要卡1，然后20和50，得分：

10\*1\*50 + 50\*20\*5 + 10\*50\*5 = 500+5000+2500 = 8000.

如果玩家的选取顺序是50 20 1，则他的得分是：

1\*50\*20 + 1\*20\*5 + 10\*1\*5 = 1000+100+50 = 1150.

**输入**

第一行输入n(3<=n<=100)表示卡片的数量，第二行输入1到n表示每张卡片上的数字。

**输出**

一行一个整数，表示最小的得分。

**样例输入**

6

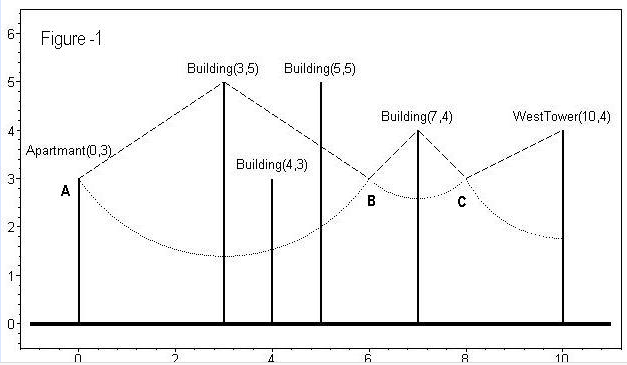
10 1 50 50 20 5

**样例输出**

3650

**Poj1925**

章鱼博士绑架了Spiderman的女友乔丹并将她放在在西塔。现在的英雄Spiderman必须尽快到达西塔就可以救她。用他自己的武器网络。从Spiderman的公寓，从那里他开始，到塔有一条路。路的旁边站着许多高大的建筑物，它肯定比他公寓更高或相等。Spiderman能把他的网搭建在任何塔和自己之间（包括塔），然后摆到大楼的另一边。在他完成秋千的那一刻，他可以把自己的网拍到另一座大楼，再拍一次，直到他到达西塔为止。图一显示了Spiderman怎样从上他的公寓–他荡塔从A到B，从B到C，从C到塔。所有的建筑物（包括塔）都被视为直线，在他荡秋千时，他不能击中地面，这意味着网络的长度较短或等于建筑物的高度。注意到在Spiderman的运动，他永远不会倒退。



你可以假设每一个摆动都需要一个时间单位。如图，Spiderman用3次到达塔，你可以很容易地发现没有更好的办法。

**输入**

输入的第一行包含了测试用例的数量（1<= K<= 20）。每一种情况开始于一个包含单一的整数n（2 < = n <= 5000），建筑物的数量（包括公寓和塔）。接着n行，每行包含两个整数，Xi、Yi，（0<=i，Yi< = 1000000）的位置和高度的建筑物。第一个建筑总是公寓,最后一个总是塔。输入按上升顺序按值排序，没有两个建筑物具有相同的x值。

**输出**

对于每一个测试案例，输出一行可行的最小数目（如果到达塔很可能），如果Spiderman不能到达塔则输出-1。

**样例输入**

2

6

0 3

3 5

4 3

5 5

7 4

10 4

3

0 3

3 4

10 4

**样例输出**

3

**Poj1837 balance**

Gigel有一个奇怪的天平，他希望修好它，事实上，这个天平与其它的天平不同。

它有两根臂，重量忽略不计，长度是15.有一些挂钩已经挂在上面了，Gigel’想要挂一些它收集到的G（1<=G<=20）件物体，每件物品重量为1..25，Gigel可以拿下来每一个挂钩，但他强迫自己必须用上所有的挂钩。

最后，Gigel设法使用在NOI中学到的经验来平衡设备。现在他希望知道有多少种方法可以平衡设备。

了解到最开始挂钩的重量与位置，写一个程序，计算平衡设备存在的方法。

题目保证存在至少一种方法可以解决。

**输入数据**

输入符合以下结构：

第一行是C（2 <= C <= 20）和G（2 <= G <= 20）；

下一行是C个整数，范围是-15..15来描述C个挂钩的位置，天平臂可看出一个X轴，中点是远点，左臂是‘-’，右臂是‘+’

接下来一行是G件物品的重量，范围是1..25。

**输出数据**

一个数M，表示平衡天平可以有M种方法

**样例输入**

2 4

-2 3

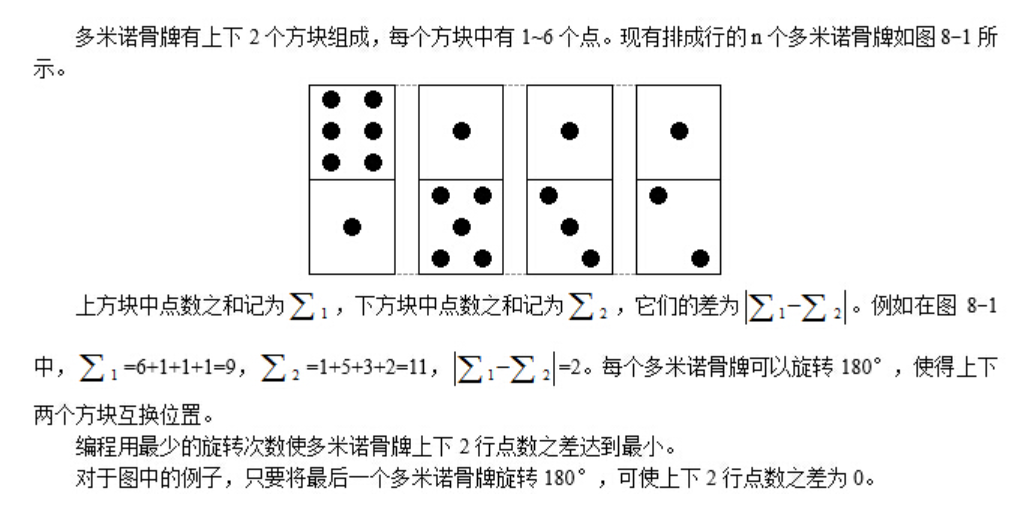
3 4 5 8

**样例输出**

2

[**http://fz.openjudge.cn/dp2/23/**](http://fz.openjudge.cn/dp2/23/) **多米诺骨牌**

**描述**



**输入**

输入文件的第一行是一个正整数n(1≤n≤1000)，表示多米诺骨牌数。接下来的n行表示n个多米诺骨牌的点数。每行有两个用空格隔开的正整数，表示多米诺骨牌上下方块中的点数a和b，且1≤a，b≤6。

**输出**

输出文件仅一行，包含一个整数。表示求得的最小旋转次数。

**样例输入**

4

6 1

1 5

1 3

1 2

**样例输出**

1

**Poj 1636 Prison rearrangement**

为了降低监狱风险，和逃脱概率，两个离得较近的监狱打算重新安排下囚犯。他们希望交换一个监狱中的一半囚犯， 。根据囚犯的犯罪史，他们知道一些囚犯放在同一个监狱是非常危险的，因此应该让他们最好分开。监狱头头们想要让这些危险的囚犯都分开囚禁，但是，他们发现，要将所有应该分开的囚犯都分开，这个事情不可能实现。因此，他们想找出一个尽可能能分开一半监狱的人的方案。

**输入**

第一行是一个单独的正整数n，是测试数据组数。

每组数据第一行是两个非负整数：m和r，1<m<200，m为两个监狱的人数，r为危险人的人的数目。接下来r行，每行一对正整数：xi,yi，都在1到m这个范围内，表示xi与yi不能在同一个监狱

**输出**

对于每组测试数据，输出一行，k<=m/2，第一个监狱最多能分k个囚犯到第二个监狱去，并且他们两两之间都不会发生危险的事情。

**样例输入：**

3

101 0

3 3

1 2

1 3

1 1

8 12

1 1

1 2

1 3

1 4

2 5

3 5

4 5

5 5

6 6

7 6

8 7

8 8

**样例输出**

50

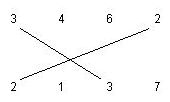
0

3

**Poj 1692 Crossed Matchings**

**描述**

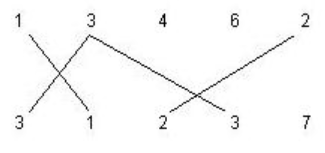
这里有两行正整数，我们能在两个相同的数字之间（如果一个数字在第一行，第二个数字在第二行）。如果匹配的数字为r，我们称这种匹配线段为r-匹配线段，例如，下图展示了2-匹配线段和3-匹配线段



我们希望找到最多的匹配数符合以下条件：

1. 每一个匹配要与另外一个且仅能和一个匹配线交叉，并且两个的匹配值不能相同

2、没有两个匹配线能连接同一个数字，例如，下图就是不被允许的。



写一个程序计算输入数据中的最大匹配数，最大匹配数字一直都会是偶数。

**输入：**

第一行数字M(1 <= M <= 10),表示测试数据组数。

每一组测试数有三行。

第一行N1，N2，分别代表第一行和第二行的数的数量。

第二行N1个正整数，表示第一行，

第二行N2个正整数，表示第二行

所有的正整数都小于100

**输出**

输出没组数据单独一行，最大匹配数

**样例输入**

3

6 6

1 3 1 3 1 3

3 1 3 1 3 1

4 4

1 1 3 3

1 1 3 3

12 11

1 2 3 3 2 4 1 5 1 3 5 10

3 1 2 3 2 4 12 1 5 5 3

**样例输出**

6

0

8