**写在前面的话：**

祝贺同学们，马上就要开学了，因此这是一套庆祝大家开学的题目（真是个好理(jie)由(kou)）。

本套题共四题，每题限制：**空间128M，时间1000ms**。考试时间：3小时20分钟（11:50结束，大家可以提前去吃饭）

为了降低难度，将会给大家算法提示，这是一套：

**搜索训练题**

当然如果能想到更好的算法，一定要推荐给大家。

**神经网络**

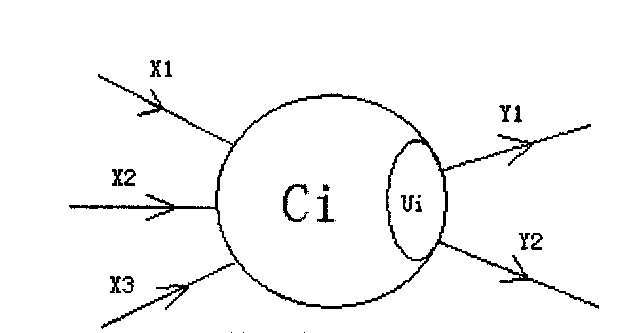
**（network）**

【问题背景】

人工神经网络（Artificial Neural Network）是一种新兴的具有自我学习能力的计算系统，在模式识别、函数逼近及贷款风险评估等诸多领域有广泛的应用。对神经网络的研究一直是当今的热门方向，兰兰同学在自学了一本神经网络的入门书籍后，提出了一个简化模型，他希望你能帮助他用程序检验这个神经网络模型的实用性。

【问题描述】

在兰兰的模型中，神经网络就是一张有向图，图中的节点称为神经元，而且两个神经

元之间至多有一条边相连，下图是一个神经元的例子：

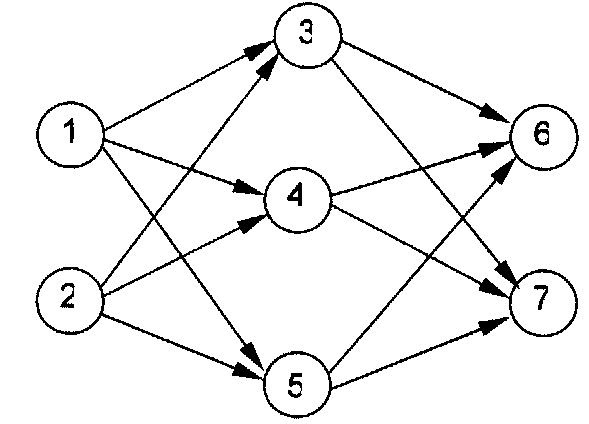
神经元〔编号为1）

图中，X1—X3是信息输入渠道，Y1－Y2是信息输出渠道，C1表示神经元目前的状态，

Ui是阈值，可视为神经元的一个内在参数。

神经元按一定的顺序排列，构成整个神经网络。在兰兰的模型之中，神经网络中的神

经元分为几层；称为输入层、输出层，和若干个中间层。每层神经元只向下一层的神经元

输出信息，只从上一层神经元接受信息。下图是一个简单的三层神经网络的例子。

兰兰规定，Ci服从公式：（其中n是网络中所有神经元的数目）



公式中的Wji（可能为负值）表示连接j号神经元和 i号神经元的边的权值。当 Ci大于0时，该神经元处于兴奋状态，否则就处于平静状态。当神经元处于兴奋状态时，下一秒

它会向其他神经元传送信号，信号的强度为Ci。

如此．在输入层神经元被激发之后，整个网络系统就在信息传输的推动下进行运作。

现在，给定一个神经网络，及当前输入层神经元的状态（Ci），要求你的程序运算出最后网

络输出层的状态。

【输入格式】

输入文件第一行是两个整数n（1≤n≤100）和p。接下来n行，每行两个整数，第i＋1行是神经元i最初状态和其阈值（Ui），非输入层的神经元开始时状态必然为0。再下面P行，每行由两个整数i，j及一个整数Wij，表示连接神经元i、j的边权值为Wij。

【输出格式】

输出文件包含若干行，每行有两个整数，分别对应一个神经元的编号，及其最后的状

态，两个整数间以空格分隔。仅输出最后状态非零的输出层神经元状态，并且按照编号由

小到大顺序输出！

若输出层的神经元最后状态均为 0，则输出 NULL。

【输入样例】

5 6

1 0

1 0

0 1

0 1

0 1

1 3 1

1 4 1

1 5 1

2 3 1

2 4 1

2 5 1

【输出样例】

3 1

4 1

5 1

**埃及分数**

**(egypt)**

**题目描述**

在古埃及，人们使用单位分数的和(形如1/a的, a是自然数)表示一切有理数。 如：2/3=1/2+1/6,但不允许2/3=1/3+1/3,因为加数中有相同的。 对于一个分数a/b,表示方法有很多种，但是哪种最好呢？ 首先，加数少的比加数多的好，其次，加数个数相同的，最小的分数越大越好。 如： 19/45=1/3 + 1/12 + 1/180 19/45=1/3 + 1/15 + 1/45 19/45=1/3 + 1/18 + 1/30, 19/45=1/4 + 1/6 + 1/180 19/45=1/5 + 1/6 + 1/18. 最好的是最后一种，因为1/18比1/180,1/45,1/30,1/180都大。 给出a,b(0<a<b<1000),编程计算最好的表达方式。

**输入描述**

a b

**输出描述**

若干个数，自小到大排列，依次是单位分数的分母。

**样例输入**

19 45

**样例输出**

5 6 18

**数据范围**

20%： 0<a<b<10

50%： 0<a<b<100

100%： 0<a<b<1000

**木棒**

**（stick）**

**题目描述**

乔治拿来一组等长的木棒，将它们随机地砍断，使得每一节木棍的长度都不超过50个长度单位。然后他又想把这些木棍恢复到为裁截前的状态，但忘记了初始时有多少木棒以及木棒的初始长度。请你设计一个程序，帮助乔治计算木棒的可能最小长度。每一节木棍的长度都用大于零的整数表示。

**输入**

输入包含多组数据，每组数据包括两行。第一行是一个不超过64的整数，表示砍断之后共有多少节木棍。第二行是截断以后，所得到的各节木棍的长度。在最后一组数据之后，是一个零。

**输出**

为每组数据，分别输出原始木棒的可能最小长度，每组数据占一行。

**样例输入**

9

5 2 1 5 2 1 5 2 1

4

1 2 3 4

0

**样例输出**

6

5

**数据范围**

20%： 组数=1,木棍数目<=10

50%： 组数<=2,木棍数目<=40

100%： 组数<=4,木棍数目<=64

聪明的打字员

(clever）

阿兰是某机密部门的打字员，她现在接到一个任务：需要在一天之内输入几百个长度固定为6的密码。当然，她希望输入的过程中敲击键盘的总次数越少越好。

不幸的是，出于保密的需要，该部门用于输入密码的键盘是特殊设计的，键盘上没有数字键，而只有以下六个键：Swap0, Swap1, Up, Down, Left, Right，为了说明这6个键的作用，我们先定义录入区的6个位置的编号，从左至右依次为1，2，3，4，5，6。下面列出每个键的作用：

Swap0：按Swap0，光标位置不变，将光标所在位置的数字与录入区的1号位置的数字（左起第一个数字）交换。如果光标已经处在录入区的1号位置，则按Swap0键之后，录入区的数字不变；

Swap1：按Swap1，光标位置不变，将光标所在位置的数字与录入区的6号位置的数字（左起第六个数字）交换。如果光标已经处在录入区的6号位置，则按Swap1键之后，录入区的数字不变；

Up：按Up，光标位置不变，将光标所在位置的数字加1（除非该数字是9）。例如，如果光标所在位置的数字为2，按Up之后，该处的数字变为3；如果该处数字为9，则按Up之后，数字不变，光标位置也不变；

Down：按Down，光标位置不变，将光标所在位置的数字减1（除非该数字是0），如果该处数字为0，则按Down之后，数字不变，光标位置也不变；

Left：按Left，光标左移一个位置，如果光标已经在录入区的1号位置（左起第一个位置）上，则光标不动；

Right：按Right，光标右移一个位置，如果光标已经在录入区的6号位置（左起第六个位置）上，则光标不动。

当然，为了使这样的键盘发挥作用，每次录入密码之前，录入区总会随机出现一个长度为6的初始密码，而且光标固定出现在1号位置上。当巧妙地使用上述六个特殊键之后，可以得到目标密码，这时光标允许停在任何一个位置。

现在，阿兰需要你的帮助，编写一个程序，求出录入一个密码需要的最少的击键次数。

**输入文件（clever.in）**

文件仅一行，含有两个长度为6的数，前者为初始密码，后者为目标密码，两个密码之间用一个空格隔开。

**输出文件（clever.out）**

文件仅一行，含有一个正整数，为最少需要的击键次数。

**输入样例**

123456 654321

**输出样例**

11

**样例说明：**

初始密码是123456，光标停在数字1上。对应上述最少击键次数的击键序列为：

|  |  |
| --- | --- |
| 击键序列： | 击键后的录入区  （下划线表示光标所在位置） |
|  | 123456 |
| Swap1 | 623451 |
| Right | 623451 |
| Swap0 | 263451 |
| Down | 253451 |
| Right | 253451 |
| Up | 254451 |
| Right | 254451 |
| Down | 254351 |
| Right | 254351 |
| Up | 254361 |
| Swap0 | 654321 |

最少的击键次数为11。