

老严的奶茶店（函数）

老严的奶茶店出了一款新品奶茶，奶茶里面加了可可粉。如果把调制好的奶茶静止放着，可可粉就会沉淀，从而影响口感。

为了迎来中午休息时段的大客流，老严提前调制了一些奶茶，分别放在了三个桶里。这些桶可能有不同的容积，也可能并没有完全装满。然后将桶1的奶茶倒入桶2，然后将桶2中的奶茶倒入桶3，然后将桶3中的奶茶倒入桶1，然后再将桶1的奶茶倒入桶2，如此周期性地操作，共计进行100次（所以第100次操作会是桶1倒入桶2），刚好迎来客流高峰。

当老严将桶a中的奶茶倒入桶b时，他会倒出尽可能多的奶茶，直到桶a被倒空或是桶b被倒满。请告诉老严当他倒了100次之后每个桶里将会有多少牛奶。

输入格式（文件名：mixmilk.in）：

输入文件的第一行包含两个空格分隔的整数：第一个桶的容积c1，以及第一个桶里的奶茶量m1。c1和m1均为正整数，并且不超过 10^9 。第二和第三行类似地包含第二和第三个桶的容积和奶茶量。

输出格式（文件名：mixmilk.out）：

输出三行，给出倒了100次之后每个桶里的奶茶量。

输入样例：

```
10 3
11 4
12 5
```

输出样例：

```
0
10
2
```

在这个例子中，每倒一次之后每个桶里的奶茶量如下：

初始状态：3 4 5

1. 桶1->2：0 7 5

2. 桶2->3：0 0 12

3. 桶3->1：10 0 2

4. 桶1->2：0 10 2

5. 桶2->3：0 0 12

（之后最后三个状态循环出现……）

```

1 #include <stdio>
2 #include <algorithm>
3 using namespace std;
4
5 void pour(int& c1, int& m1, int& c2, int& m2) {
6     int amt = min(m1, c2 - m2);
7     m1 -= amt;
8     m2 += amt;
9 }
10
11 int main() {
12     int c1, c2, c3;
13     int m1, m2, m3;
14     scanf("%d %d", &c1, &m1);
15     scanf("%d %d", &c2, &m2);
16     scanf("%d %d", &c3, &m3);
17
18     for (int i = 0; i < 33; i++) {
19         pour(c1, m1, c2, m2);
20         pour(c2, m2, c3, m3);
21         pour(c3, m3, c1, m1);
22     }
23     pour(c1, m1, c2, m2);
24
25     printf("%d\n%d\n%d\n", m1, m2, m3);
26 }

```

实力对比

为了参加区赛，四眼碛小学的小伙伴们开始进行编程训练了！Lao Yan 选定了他最喜爱 N 个同学参加训练，同时评估他们学习不同的编程技术的进度。

总共进行了 K 次训练课（ $1 \leq K \leq 10$ ），每次课后Lao Yan 都会根据 N 个同学的表现给他们进行排名（ $1 \leq N \leq 20$ ）。之后，Lao Yan对这些排名的一致性产生了好奇。

对于两个不同的同学，怎么样实力值算是*一致的呢*？如果其中一个同学在每次训练课中都表现得都比另一个要好。

请帮助Lao Yan计算实力对比一致的同学的对数。

输入格式（文件名：gymnastics.in）：

输入的第一行包含两个正整数 K 和 N 。以下 K 行每行包含整数 $1 \dots N$ 的某种排列，表示同学们的排名（同学们用编号 $1 \dots N$ 进行区分）。如果在某一行中 A 出现在 B 之前，表示同学 A 表现得比同学 B 要好。

输出格式（文件名：gymnastics.out）：

输出一行，包含实力对比一致的同学的对数。

输入样例：

```
3 4
4 1 2 3
4 1 3 2
```

输出样例：

```
4
实力对比一致的同学对为 (1,4)、(2,4)、(3,4)和 (1,3)。
```

```

1 #include <iostream>
2 #include <fstream>
3 using namespace std;
4
5 int N, K;
6 int data[10][20];
7
8 bool better(int a, int b, int session)
9 {
10     int apos, bpos;
11     for (int i=0; i<N; i++) {
12         if (data[session][i] == a) apos = i;
13         if (data[session][i] == b) bpos = i;
14     }
15     return apos < bpos;
16 }
17
18 int Nbetter(int a, int b)
19 {
20     int total = 0;
21     for (int session=0; session<K; session++)
22         if (better(a,b,session)) total++;
23     return total;
24 }
25
26 int main(void)

```

```

26 int main(void)
27 {
28     ifstream fin ("gymnastics.in");
29     ofstream fout ("gymnastics.out");
30     fin >> K >> N;
31     for (int k=0; k<K; k++)
32         for (int n=0; n<N; n++)
33             fin >> data[k][n];
34     int answer = 0;
35     for (int a=1; a<=N; a++)
36         for (int b=1; b<=N; b++)
37             if (Nbetter(a,b) == K) answer++;
38     fout << answer << "\n";
39     return 0;
40 }

```

分析：

if (Nbetter(a,b) == K) answer++;

a和b比赛，如果胜了k次，那么答案就加一个