



题目名称	探险队	突破塔门	升降梯口
程序文件名	explore	flower	elevator
输入文件名	explore.in	flower.in	elevator.in
输出文件名	explore.out	flower.out	elevator.out
每个测试点时限	1 秒	1 秒	1 秒
内存限制	128 MB	128 MB	128 MB
测试点数目	10	10	10
每个测试点分值	10	10	10
是否有部分分	无	无	无
评测方式	Normal	Special Judge	Normal

提交源程序需加后缀

对于 Pascal 语言	explore.pas	flower.pas	elevator.pas
对于 C 语言	explore.c	flower.c	elevator.c
对于 C++ 语言	explore.cpp	flower.cpp	elevator.cpp

评测环境:

Intel(R) Core(TM) i3-370M CPU @2.40GHz 2.39GHz, 2.00GB RAM

Cena 0.8.2 @ Windows 8 Enterprise RTM x64

C++选手注意 Windows 7/8/Vista 可以使用 %lld 或 %I64d 输入输出 64 位整数。

最终评测时，所有编译命令均不打开任何优化开关。

Nescafe 27

探险队 (explore.pas/c/cpp)

题目描述

N 个探险家组队去探索神秘的 Nescafe 之塔。在路上有个记者问每个探险家参加过的探险活动的数目，而探险家们都不愿意告诉他。于是这个记者得到的回答都是“有 A 个探险家参过的探险活动比我多，而 B 个探险家参加过的探险活动比我少。”当然并不一定所有的探险家都说了实话。现在这个杯具的记者想请你帮助他计算至少有多少个探险家说了谎？

输入格式

第一行是一个正整数 N ，表示探险队的人数。

之后 N 行每行两个整数 A 和 B ，表示一个探险家的答案。

输出格式

输出一个整数表示答案，即至少有多少人说了谎。

样例输入

```
3
2 0
0 2
2 2
```

样例输出

```
1
```

数据范围与约定

对于 30% 的数据，满足 $1 \leq N \leq 1000$ 。

对于 100% 的数据，满足 $1 \leq A, B \leq N \leq 100000$ 。

Nescafe 27

突破塔门 (dash.pas/c/cpp)

题目描述

历经长途跋涉之后，探险队员们终于来到了 Nescafe 之塔的一层大殿门前。而要想进入 Nescafe 之塔，就必须完成塔门上的一道解谜游戏，从而突破紧闭的石门。这个任务被交给队员 Alice 完成。

这个名为《Flower》的解谜游戏一共有 n 关。该游戏十分奇怪，不管有没有成功解开某一关的谜题都可以选择直接进入下一关，但游戏通关后会给出一个评分，评分的计算方式为所有连续成功解谜的关卡数的 k 次方之和，例如：若用 O 表示成功解谜，X 表示直接跳过，则在 n 为 5， k 为 2，Alice 解谜情况为 OXOOX 的情况下，她能得到的分数为 $1*1+2*2=5$ 。

为了兼顾通关速度和最终获得的分数，Alice 决定若在超过 233 秒后她仍没有解出某关的谜题，就直接跳过这一关。现在，Alice 的朋友穆秋帮助她算出了 Alice 在 233 秒内解出某关谜题的概率，你需要做的便是计算出 Alice 通关时的期望得分。

经历了梭哈事件和蹦极事件的 Alice 对现代科技已经彻底丧失信心，因此你的结果不必特别精确就能让她满意。

输入格式

第一行包含两个正整数 n, k ，分别表示游戏有 n 关，计分方式为连续关数的 k 次方。

接下来的一行中有 n 个 0 至 1 之间的实数，表示 Alice 在限定时间内通过某关的概率。

输出格式

输出一个实数，表示 Alice 通关时的期望得分。你可以输出任意位数的实数，也可以输出采用科学计数法的数字，但需要保证你输出的答案与标准答案的相对误差小于 eps 。（对于 $k=1$ 或 $k=2$ ， eps 为 $1e-9$ ，对于 $k=3$ 或 $k=4$ ， eps 为 $1e-5$ ）。

样例输入

```
3 2
0.5 0.5 0.5
```

样例输出

```
2.75
```

数据范围与约定

测试点编号	N	K	说明
0	20	2	-
1	300	2	-

Nescafe 27

2	1000	2	-
3	100000	2	-
4	100000	1	-
5	1000	4	-
6	20000	3	输入数据中所有实数小数部分均不超过 3 位。
7	50000	4	输入数据中所有实数小数部分均不超过 2 位。
8	80000	3	输入数据中所有实数小数部分均不超过 3 位。
9	100000	3	输入数据中所有实数小数部分均不超过 3 位。

标准答案由高精度计算得出，你的结果不需要与其完全相同，但应符合输出格式中的精度要求。

提示

C++选手如果使用 `cin` 读入数据很可能因此超时，推荐使用 `scanf/printf`。

最终评测环境支持 80 位浮点数数据类型，即 C/C++ 中的 `long double` 和 Pascal 中的 `extended`。

Nescafe 27

升降梯口 (elevator.pas/c/cpp)

题目描述

探险队员们终于突破了塔门，来到了 Nescafe 之塔的一层大殿。而探险队员们想要拜访的四大护法和塔之圣主据传都住在塔的高层，因此他们迅速奔向了大殿一侧的升降梯口。不过由于 Nescafe 之塔乃是一年一度封印和开启“Nescafe 之杯”的禁地，自然不能允许外来人员随意闯入，因此要想乘坐升降梯，队员们必须破解开启升降梯动力的密码。

升降梯的密码盘是一个由 $n \times n$ 个方格组成的正方形(n 为偶数)，第 i 行第 j 列的方格中标着数字 $(i-1) \times n + j$ ，而在密码盘的上面覆盖着一张同样由 $n \times n$ 个方格组成的挡光片。如果挖去挡光片上的 $n \times n / 4$ 个格子，并从小到大记下通过挖去的格子看到的 $n \times n / 4$ 个数；然后把挡光片顺时针旋转 90° 、 180° 、 270° ，每次也同样记下看到的 $n \times n / 4$ 个数；这样最终将得到 $n \times n$ 个数。如果记下的这 $n \times n$ 个数恰好是 $1 \sim n \times n$ 的一个排列，那么我们称这片挖去 $n \times n / 4$ 个格子后的挡光片对密码盘是“精确覆盖”的。不妨用一个 $n \times n$ 的 01 矩阵表示这张挡光片，其中挖去的格子为 1，没有挖去的格子为 0，那么如下图所示的挡光片就可以表示为：

1	2	3	4	0000
5	6	7	8	0000
9	10	11	12	1101
13	14	15	16	0001

定义挡光片 A 比挡光片 B 小，当且仅当 A 对应的 01 矩阵的字典序小于 B 对应的 01 矩阵的字典序（即：A 和 B 对应的 01 矩阵中存在一个位置 p ，使得矩阵 A、B 中前 $p-1$ 个数字相同，而 A 中第 p 个数字为 0，B 中第 p 个数字为 1）。

现在升降梯口的墙上写着一个数字 k ，探险队员们必须迅速制作出第 k 小的、对密码盘“精确覆盖”的挡光片，用以在密码盘上获取 $n \times n$ 个数作为开启升降梯动力的密码。

输入格式

一行两个正整数 n 、 k 。

输出格式

输出满足要求的挡光片对应的 01 矩阵。

样例输入

4 15

样例输出

0000

Nescafe 27

0000
1101
0001

数据范围与约定

测试点编号	n	k
#1	=2	<=10
#2	=2	<=10^3
#3	=4	<=10
#4	=4	<=10^9
#5	=6	<=10
#6	=6	<=10^18
#7	=8	<=10
#8	=8	<=10^18
#9	=10	<=10^18
#10	=10	<=10^18