

# “华为杯” 2015 年南京高校程序设计邀请赛 暨第十二届南京大学程序设计竞赛

## 试题册

2015 年 4 月 26 日

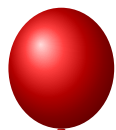
13:00 – 17:00

### Contents

A A Check-in Problem 签到题	2
B Big News 大新闻	4
C Containing 包含关系	6
D Deviation 标准差	8
E Excited!	10
F Faster than Anyone Else 跑得比谁都快	12
G Game 棋盘游戏	16
H Hotpot 麻辣香锅	18



所有题目均为标准输入输出；时间、内存限制以 Online Judge 为准。



## A A Check-in Problem 签到题

### 题目描述

又到了一年一度南京大学程序设计竞赛的时候了！欢迎各位来自兄弟院校的朋友和中学生小朋友们！为了让每个同学都能开心地拿着起球离开赛场，主办方一定会给大家准备一道签到题，今天的签到题又刚好是处理签到。

故事要从这次负责命题的 zcl 说起。zcl 作为一个看机房 (端茶倒水) 的少年，正在为选手签到的事情困扰着。zcl 有着一张机房进出表，每位进出机房的选手都会来找 zcl 登记。zcl 会在表格上给每位选手登记一个记录，整个表格记录若干行，每行都是 (名字, 学校, 出/进) 的形式，其中名字为选手名，学校为选手所属的学校名字，出/进代表这位选手是出机房门还是进机房门。例如：

名字	学校	出/进
zcl	Nanjing University	进
zcl	Nanjing University	出
zck	Southeast University	进
...	...	...

现在由于 zcl 的疏忽，表格的一些地方沾上了墨水，导致一些进出记录看不清了。好在 zcl 很清楚地知道他对所有选手的进出都做了登记，因此同一名选手在列表上先后的两条记录一定满足以下性质：

1. 没有进机房前不能出机房。
2. 不能在出机房之后再次出机房。
3. 不能在进机房之后再次进机房。

由于表格可能很长 (多达 100,000 次记录)，现在 zcl 想请你编程帮他恢复这个表格，聪明的你能帮助他吗？

### 输入格式

数据第一行包含一个整数  $n$ ，表示这个表格一共有  $n$  行。接下来  $n$  行描述了 zcl 手中的表格，每一行的格式如下：

选手名字 选手学校 选手进/出

其中，选手的名称与学校均只包含小写字母。选手名字与选手学校、选手进/出之间以一个空格隔开。选手进用一个大写字母 'I' 表示，选手出用大写字母 'O' 表示，zcl 不知道用星号 '\*' 表示 (均不包括引号)。

## 输出格式

对于每组数据，输出  $n$  行，表示恢复过的表格，即将输入中所有的 '\*' 用 'I' 或 'O' 替换，并满足之前的三个性质。

## 样例数据

样例输入 1	样例输出 1
3 zcl nju * zcl nju * zck seu I	zcl nju I zcl nju O zck seu I

## 数据说明

注意来自不同学校的选手可能重名，但他们是不同的选手。在表格的两行里，名字和学校都相同的选手则认为是同一个选手 (即同一个学校没有重名的选手)。

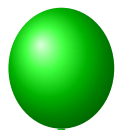
对于公开组，数据满足  $1 \leq n \leq 100$ 。

对于提高组，数据满足  $1 \leq n \leq 10,000$ 。

所有数据均保证  $1 \leq \text{选手名字长度} \leq 20$ ,  $1 \leq \text{选手名字长度} \leq 20$ ，且存在唯一合法的输出表格。

## 提示

输入文件可能很大，因此请使用高效的输入输出方式 (例如 `scanf`, `printf` 和 `BufferedReader`)。



## B Big News 大新闻

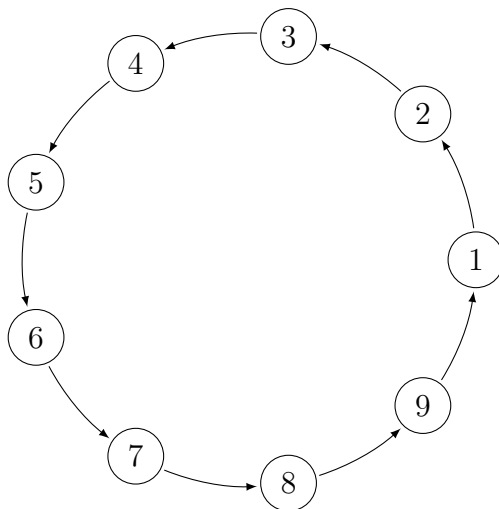
### 题目描述

水果世界里有  $n$  个城市，每个城市每天都在发生大新闻。

大菠萝是一位新闻工作者，它要负责管理和指挥小菠萝们在水果世界的各个城市之间传播各地发生的大新闻。

水果世界的科技水平是远远不如人类世界的，水果们主要使用一种古老的特制发报机传递信息。每台城市都有一台这样的发报机，一台发报机可以接收其它发报机向它发送的信息，同时还可以指定一个城市，向这个城市的发报机发出信息。每个城市都有一个小菠萝负责操作发报机，小菠萝会把当地发生的大新闻以及当地发报机接收到的其他城市的大新闻，发送到这台发报机指定的城市。

一开始，每个城市的发报机都已经指定了一个城市进行发送。有一天，大菠萝发现一些城市发生的大新闻不能传播到另一些城市，它决定对发报机们的指定城市进行修改，使得任意城市的大新闻都能传播到其他所有城市。由于每个城市只能向另一个城市传播新闻，最终调整后所有城市的发报系统一定连成一个环：



当然，修改发报系统的代价是很大的，所以大菠萝希望修改的发报机数量越少越好。大菠萝希望你能给出一个方案，使得在修改的发报机数量最少的情况下，让任一城市的大新闻都能传播到所有的城市。

### 输入格式

输入有 2 行。

第一行只有一个正整数  $n$ ，代表城市的数量。

第二行有  $n$  个正整数  $a_i$  ( $1 \leq a_i \leq n$ )，代表城市  $i$  的发报机指定了城市  $a_i$  进行发送。相邻的 2 个数用一个空格隔开，行末没有空格。

需要注意的是，有些发报机一开始可能会发送给自己，因为若干年前设置这些发报机的老菠萝老眼昏花了。

## 输出格式

输出只有 1 行，一共  $n$  个正整数  $b_i$  ( $1 \leq b_i \leq n$ )，代表修改完毕后城市  $i$  的发报机的指定城市为城市  $b_i$ ，相邻的 2 个数用一个空格隔开，行末没有空格。注意你必须输出一个合法的最优方案：任意城市的大新闻都能传播到其他所有城市，并且修改的发报机数量最少。

## 样例数据

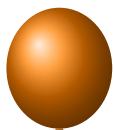
样例输入 1	样例输出 1
3 2 1 1	2 3 1
样例输入 2	样例输出 2
4 2 4 1 3	2 4 1 3

## 数据说明

对于公开组，数据满足  $1 \leq n \leq 233$ 。

对于提高组，数据满足  $1 \leq n \leq 233,333$ 。

本题采用 Special Judge。任何合法的输出都会被接受。



## C Containing 包含关系

### 题目描述

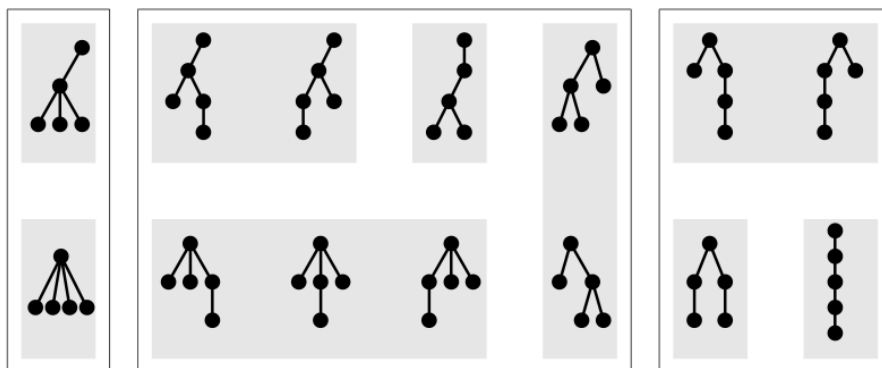
Adder 打算编写一个名为 `advAdder` 的 AI 程序来代替自己工作 (比如帮他出题)。为了让 `advAdder` 能做尽量多的事情, Adder 首先需要教会 AI 大量的知识。

Adder 认为所有的知识都可以拆成实体和实体上的关系, 比如 “Adder 是个傻瓜” 就可以看成从 “Adder” 实体到 “傻瓜” 实体有一个标记为 “是” 的有向关系。Adder 发现一些复杂的知识可能需要把许多这样的关系复合起来, 比如 “`advAdder` 这样的程序比 Adder 这样的人类更有用” 包括以下三个关系:

1. (“`advAdder`”, 是, “程序”);
2. (“Adder”, 是, “人类”);
3. (“`advAdder`”, 比 `xx` 更有用, “Adder” )。

Adder 在构造知识的过程中, 发现有很多知识都是树状的结构, 而且其中中出现了一些共同的结构。因此, Adder 猜测有一些结构是被广泛的使用在知识的表示之中的, 并且决定自己设计一些模式在构造出的知识树上验证一下。为了减轻你的负担, Adder 会直接给你知识树的抽象结构, 你只要输出判定的结果即可。

具体来说, Adder 希望你来替他判断他猜测的一个结构是否被给出的知识树中包含了, 即验证给定的一个无向无根树 (结构树  $U$ ) 是否被包含在另一个无向无根树 (知识树  $T$ ) 中。换言之, 即判定是否存在知识树  $T$  的一棵子树  $V$  与  $U$  同构 (在忽略结点编号之后是同一棵无根树)。下图列举了 5 个结点树的同构关系。虽然树的结构千变万化, 但总共只有三种同构意义下不同的无根树。



### 输入格式

本题一个输入包含多组数据。输入第一行一个  $T$  表示输入数据组数。

每组数据第一行有两个整数  $S$  和  $N$  分别表示 Adder 猜测的结构树  $U$  和知识树  $T$  的大小。

接下来  $S - 1$  行每行两个小写字母  $a, b$  ( $a, b$  均在前  $S$  个小写字母中), 表示结构树上存在一个  $a$  到  $b$  的无向关系。

接下来  $N - 1$  行每行两个数字  $u, v$  ( $0 \leq u, v < T$ ), 表示知识树上存在一个  $u$  到  $v$  的无向关系。

输出格式

每组数据输出一行, 如果结构树是知识树的一个子树, 输出 “Contain”, 否则输出 “Not contain” (均不包含引号)。

样例数据

样例输入 1	样例输出 1
1 4 10 a b a c a d 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9	Contain

数据说明

所有数据满足  $1 \leq T \leq 20$ 。  
对于公开组和提高组, 数据均满足  $4 \leq S \leq 26, S \leq N \leq 200$ 。



## D Deviation 标准差

### 题目描述

Adder 编写了一个名为 advAdder 的 AI 程序来代替自己工作，现在 Adder 想知道 advAdder 的计算能力是否可靠，决定拿一些计算问题来考一考它。微积分没有学好的 Adder 能够想到的最合适常见问题就是求一个序列的标准差这个加减乘除平方开根都有的问题。

具体来说，给定数值  $x_1, x_2, \dots, x_n$ ，其平均值定义为  $\mu = \frac{1}{n} \sum_{1 \leq i \leq n} x_i$ ，标准差定义

为  $\sigma = \sqrt{\frac{1}{n} \sum_{1 \leq i \leq n} (x_i - \mu)^2}$ 。

但这个问题似乎太简单了。因此 Adder 增加了一些难度：在询问标准差的过程中还会改变一些数字。现在愚蠢的 Adder 发现自己忘了写一个新的程序来验证得到的结果是否正确，他希望你能帮助他完成这个程序。

### 输入格式

本题一个输入包含多组数据。输入第一行一个  $T$  表示输入数据组数。

每组数据第一行两个数  $n, m$ ，分别表示序列的长度和询问/修改的操作总数（初始时刻序列中全为 0）。

接下来  $m$  行，每行是  $Q\ l\ r$  或者  $M\ l\ r\ v$  的形式 ( $1 \leq l \leq r \leq n, 0 \leq v \leq 10,000$ )。首字母为  $Q$  表示接下来 Adder 要查询  $[l, r]$  区间（含端点）的标准差，首字母为  $M$  表示 Adder 要把  $[l, r]$  区间的数字都改成  $v$ 。

### 输出格式

对每个  $Q$  询问输出一行，内容为保留两位小数的查询结果，表示对应区间的标准差。



## 样例数据

样例输入 1	样例输出 1
1 100 7 M 1 80 5 Q 1 100 Q 80 80 M 70 90 20 M 30 75 10000 Q 20 30 Q 80 100	  2.00 0.00 2873.36 9.99   

## 数据说明

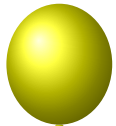
所有数据满足  $1 \leq T \leq 5$ 。

对于公开组，数据满足  $1 \leq n \leq 100,000$  且  $m \leq 1,000$ 。

对于提高组，数据满足  $1 \leq n \leq 100,000$  且  $m \leq 100,000$ 。

## 提示

输入文件可能很大，因此请使用高效的输入输出方式（例如 `scanf`, `printf` 和 `BufferedReader`）。



## E Excited!

### 题目描述

菠萝非常喜欢数字，而 (dan) 且 (shi) 他是个大学霸 (zha)。世界上有很多特殊的数字，菠萝每次见到特殊的数字都会感到非常兴奋！

有一天，无聊的菠萝发明了一个函数，并把它命名为 pineapple 函数。它认为任意一个正整数  $k$  都有一个对应的 pineapple 值，这个函数值是通过以下代码计算的：

```
function pineapple( $k$ )
begin
     $i \leftarrow 2$ ;
    while  $i \leq k$  do
        if  $k \equiv 0 \pmod{i}$  then
            return  $k/i$ ;
         $i \leftarrow i + 1$ ;
    return 1;
```

在这个函数中， $k$  可以是一个任意的正整数，并不局限于 int 或者 long long 类型，它可以很小，也可以非常非常大，甚至是接近无限大！

菠萝发现这个函数非常特殊，因为它发现对于每个正整数  $a$ ，每当它找到一个新的正整数  $b$  使得  $\text{pineapple}(b) = a$  的时候，它对  $a$  这个数字的兴奋程度又加深了一分！

菠萝把一个数的兴奋度用来表示它对这个数字的兴奋程度，一个正整数  $k$  的兴奋度的定义为所有正整数中 pineapple 值等于  $k$  的正整数的个数。比如菠萝发现有 2 个正整数 6 和 9，满足  $\text{pineapple}(6) = \text{pineapple}(9) = 3$ ，那么菠萝认为 3 这个数字的兴奋度为 2，因为它找不到除了 6 和 9 以外，pineapple 值等于 3 的正整数了。而 1 的兴奋度是无穷大的，因为菠萝发现了无数个 pineapple 值等于 1 的正整数！

菠萝想知道，如果给定一个正整数  $n$ ，它能不能分别计算出 1 到  $n$  这  $n$  个数字的兴奋度。

但是他实在太学霸 (zha) 了，以至于他根本不屑 (neng) 做这么简 (fu) 单 (za) 的计算。所以它把这个任务交给你了，然后跑去美国圣地亚哥给自己买金坷垃吃了。

### 输入格式

输入有一行，只有一个正整数  $n$ 。

## 输出格式

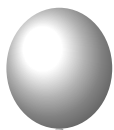
输出一共有  $n$  行，第  $i$  ( $1 \leq i \leq n$ ) 行代表  $i$  这个数字的兴奋度。如果  $i$  的兴奋度是一个有限的整数，那么输出  $i$  的兴奋度，否则输出 “Excited!” (不包含引号)，因为菠萝发现了一个能让它无限兴奋的数字！

## 样例数据

样例输入 1	样例输出 1
3	Excited! 1 2

## 数据说明

对于公开组和提高组，数据均满足  $1 \leq n \leq 2,333,333$ 。提交程序的大小不得超过 64KB。



## F Faster than Anyone Else 跑得比谁都快

### 题目描述

岛风是一只萌萌哒驱 (xiao) 逐 (xue) 舰 (sheng)。据说她有 41 节的航速，跑得比谁都快！

这一天，提督派出了岛风、吹雪以及傻级…… 呸，金刚级四姐妹金刚、比叻、榛名和雾岛组成舰队出击南方海域。然而岛风失踪了一早上才被金刚姐妹无意中用午茶点心引诱了出来，导致她们出击时不得不为了赶时间全速前进。而在出击后不久，吹雪和金刚四姐妹很快出现了一脸黑线的表情，岛风的速度实在太快了，即使是身为高速战舰的金刚级在速度上也被甩了好几个档次——当她们在全速向南前进的时候，岛风不仅一边在她们的前进方向上不落于她们，还一边不停地在左右方向上移动……



给定两个整数  $n$  和  $m$ ，我们可以把她们出击经过的整个大海域看做一个  $n$  行、 $2m + 1$  列的格子矩阵，每个格子代表一小块正方形海域，坐标记为  $(x, y)$ ，其中  $x, y$  分别为格子所在的行和列。矩阵按从北到南为第 1 到第  $n$  行，从西到东为第  $-m$  到第  $m$  列，其中有  $k$  个格子内有礁石不能驶入。我们把舰娘的移动视为从一个格子走到东、南、西 3 个方向的相邻格子中的一个（因为北边是海岸，她们是不会向北移动的）。舰队的出击航线一般是一条  $(1, 0)$  到  $(n, 0)$  的直线路径，这条线上的格子是没有礁石的，她们会停在  $(n, 0)$  这个地方跟南方的敌舰队作战。

例如，我们假设这片海域满足  $n = 6, m = 2, k = 2$ ，并且  $(2, -1)$  和  $(6, 2)$  处各有一块礁石。我们用点 (.) 代表可以驶入的格子，星号 (\*) 代表有礁石的格子，小写字母 s 代表航线的起点，小写字母 e 代表航线的终点，那么这个海域是长这样的：

```

. . s . .
. * . . .
. . . . .
. . . . .
. . . . .
. . e . *
```

我们用大于号 ( $>$ )、小于号 ( $<$ ) 和小写字母 v 分别表示舰娘在每个格子时的前进方向分别为东、南、西。终点仍然用 e 表示，因为她们会停在终点。吹雪和金刚

姐妹是严格按原定航线走的，所以她们的航线一直都是长这样的：

```

. . S . .
. * v . .
. . v . .
. . v . .
. . v . .
. . e . *
```

在多次以同样的舰队组成出击后，吹雪发现，作为一只性格非常自由散漫的舰娘，岛风总是爱走跟她们不一样的航线，而且很爱展现她引以为傲的速度。其他舰娘从北向南前进的时候，岛风在东西方向上保持与她们平行的同时，可以在【其他舰娘很努力地仅仅向南前进了一个格子】的短短时间里，不仅同样向南前进了一个格子，还向东或者向西平移了若干个格子。但是，岛风不会重复经过同一个格子，也不会经过有礁石的格子，而且舰娘是不可能飞越过礁石的。岛风也不会驶出这个矩阵之外，因为她不能离舰队太远。

简单地说，岛风的航线满足以下四个条件：

1. 岛风的移动方向只有东、南、西 3 个方向；
2. 岛风的航线不会重复经过同一个格子；
3. 岛风的航线不会经过有礁石的格子，也不会越过去；
4. 岛风的航线经过的所有格子始终保持在矩阵内。

于是吹雪发现岛风的航线实际上可能会长成左边这个样子，也可能长成中间这个样子，还可能长成右边这个样子：

```

. . > > v      v < < . .      . . v . .
. * . v <      v * . . .      . * v . .
v < < < .      > > > > v      . v < . .
v . . . .      . v < < <      . > > v .
> > > v .      v < . . .      . . v < .
. . e < *      > > e . *      . . e . *
```

让吹雪感到非常崩溃的是，尽管她们已经在这个海域一起出击很多次了，岛风的航线没有一次是相同的！这简直是在花式秀速度。吹雪很好奇，在这个海域岛风究竟有多少种不同的航行路线可以从起点走到终点。

吹雪想搞清楚这个问题，但是吹雪跟岛风同为小学生，数学水平并不好，而她们舰队另外四位的智商更加指望不上了。（这四位可是被称为傻级的舰娘啊……）于是她来求助于你，她觉得人类的数学显然要比舰娘好多了。由于这个数字可能太大，吹雪只要你求出答案对 1,000,000,007 取模就行了。

如果你帮她解决了这个问题，她会送你一个萌萌哒的 Accepted 作为感谢哟<sup>1</sup>～

## 输入格式

输入有  $k + 1$  行。第一行为 3 个整数  $n, m, k$ ，其定义见题目描述，两个整数之间用一个空格隔开。

接下来  $k$  行，每行有两个整数  $x, y$  ( $1 \leq x \leq n, -m \leq y \leq m$ )，代表  $(x, y)$  处的格子里有礁石，每两个整数之间用一个空格隔开。

数据保证吹雪和金刚姐妹的航线上没有礁石 (即不会出现  $y = 0$ ，这可是原定航线啊)，每对  $(x, y)$  不会重复出现。

## 输出格式

输出只有一行，一个整数，表示岛风的航行路线的种数对 1,000,000,007 取模的值。

## 样例数据

样例输入 1	样例输出 1
3 2 2 2 -1 2 2	5
样例输入 2	样例输出 2
3 0 0	1
样例输入 3	样例输出 3
6 2 2 2 -1 6 2	1000

## 样例说明

样例 1 中岛风有以下 5 条航线可走：

---

<sup>1</sup>感谢刚晋同学制作了这张萌萌哒的图片～



```

. . v . . . . v . . . . > v . . . . > v . v < < . .
. * v . * , . * > v * , . * v < * , . * . v * , v * . . *
. . e . . . . e < . . . . e . . . . e < . > > e . .

```

样例 2 中矩阵宽度只有 1，只要向东或者向西就会走出矩阵，于是岛风只能乖乖跟着其他舰娘一起走同一条航线了.....

样例 3 就是题目中举的例子，正好有 1000 种，但是太多了，你自己数吧 →\_→

我们认为两条航线不相同，当且仅当矩阵中存在一个格子，其中一条航线通过这个格子，另一条不通过。

// 题目背景参考动画《舰队 Collection》第 4 集，我家大傻真是太可爱啦 >.< 我家大傻才不会傻到放着对面战舰不打整天打残血驱逐舰呢（才怪.....

## 数据说明

对于公开组，数据满足  $1 \leq n \leq 233, 0 \leq m \leq 23, 0 \leq k \leq 2, 333$ 。

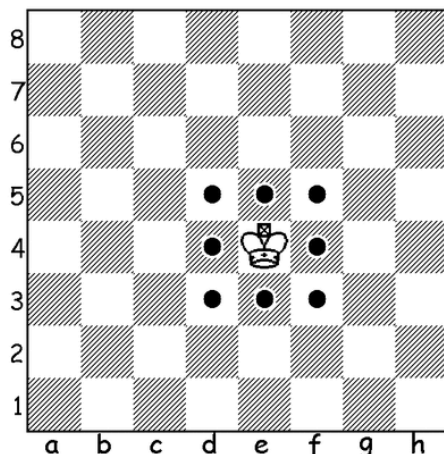
对于提高组，数据满足  $1 \leq n \leq 233, 333, 333, 0 \leq m \leq 233, 333, 333, 0 \leq k \leq 233, 333$ 。



## G Game 棋盘游戏

### 题目描述

某日, dwt 心血来潮, 突然在  $n \times n$  大小的国际象棋棋盘上, 玩起了游戏。dwt 在棋盘上放了  $n$  个王, 由于王好战的个性, 每个王将会攻击与它相邻格子里的王 (两个王相邻, 是指他们有着共同的边或者是角), 如下图所示。



无聊的 dwt 想知道总共有多少种排法, 使得 (1) 每行都恰好有一个王; (2) 每列都恰好有一个王; (3) 任意两个王互不攻击。由于方案数可能会很大, dwt 只需要你输出方案数对 1,000,000,007 取模的数值就好。

### 输入格式

输入第一行包含一个整数  $T$ , 为输入数据的组数。接下来  $T$  行, 每行包含一个整数  $n$ , 表示棋盘为  $n \times n$  大小。

### 输出格式

输出包含  $T$  行。每行包含一个整数, 为方案数对 1,000,000,007 取模的值。



## 样例数据

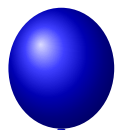
样例输入 1	样例输出 1
4	1
1	0
2	0
3	711794305
1000	

## 数据说明

对于公开组，数据满足  $1 \leq n \leq 10$ 。

对于提高组，数据满足  $1 \leq n \leq 1,000$ 。

麻麻才不会告诉你出题人本来想出  $n \leq 10,000,000$ ，但被善良的 jyy 阻止了。



## H Hotpot 麻辣香锅

### 题目描述

比赛终于顺利结束啦！好开心，ACM 队的同学们打算去吃麻辣香锅庆祝。麻辣香锅是大家非常喜爱的菜肴，可以认为是一种离线版的火锅：事先点好一些入锅的菜（例如牛肉、羊肉、五花肉等），店家将这些点好的菜加工到干锅中，一次性呈上。



然而，吃饭时总有一些战斗力为 5 的渣渣，吃一点就饱了，导致点菜的时候估计失败，临近结束时还会剩下不少没有吃完的菜，于是还有战斗力的人打算把香锅里的肉吃掉，蔬菜留下。但是你们都懂的，在一个大干锅里找肉片是个非常困难的事情！

假设香锅里有  $n$  个固形物，其中  $k$  个是肉类。假设每次一个人都是从干锅里随机夹一个固形物（每次夹都是独立事件，并且干锅内每一个固形物以等概率被夹中），如果是肉就吃掉，是蔬菜就放回去。当然我们不希望这个清扫残渣的过程持续太久，因此请你求出期望要多夹几次才能把干锅里所有的肉都吃掉。

### 输入格式

输入第一行包含一个整数  $T$ ，为输入数据的组数。接下来  $T$  行，每行包含两个整数  $n, k$  ( $k \leq n$ )，表示锅中固形物的总数，和其中属于肉类的数量。

### 输出格式

输出包含  $T$  行。每行包含一个整数，表示期望夹的次数。

### 样例数据

样例输入 1	样例输出 1
1 10 3	15.8333333333

### 数据说明

对于所有数据满足  $T \leq 10$ 。

对于公开组，数据满足  $1 \leq n \leq 100$ 。

对于提高组，数据满足  $1 \leq n \leq 100,000$ 。

本题采用 Special Judge。输出数据可保留任意位小数。只要输出与标准答案相差不超过  $10^{-4}$  即判断为正确。