

## 题目 A. Maximum Element In A Stack

现在有一个栈，初始为空。接下来有若干次操作，每次可能向栈顶 `push` 一个正整数，也可能 `pop` 掉栈顶元素。

你需要在每次操作之后计算出栈内所有元素的最大值。如果栈为空则认为此时最大值是 0。

为了避免输入文件过大，所有操作会使用 `rng61` 算法生成。同时为了避免输出文件过大，你只需要输出一个数：表示每次操作之后的答案与下标乘积的异或和。

### 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数。

接下来依次描述  $T$  组测试数据，为了减少读入量，采用如下的方式来给出数据。

对于每组测试数据：一行给出了 7 个整数依次为  $n, p, q, m, SA, SB, SC$ 。其中  $n$  表示总的操作次数，你的程序可以使用如下所示的 C++ 代码生成所有的操作。

（代码见原题面）

在这份代码中，`PUSH(v)` 表示向栈中插入一个新的元素，它的值为  $v$ 。`POP()` 表示弹出栈顶元素；如果当前栈为空，则什么也不操作（即：没有东西可以弹出）。

### 输出格式

对于每组测试数据，输出一行信息 "`Case #x: y`"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  等于  $\bigoplus_{i=1}^n (i \cdot a_i)$ ，其中  $a_i$  表示第  $i$  次操作后的答案， $\oplus$  表示异或和。

## 题目 B. Rolling The Polygon

按照逆时针绕向给出一个凸多边形的  $n$  个顶点  $P_0, P_1, \dots, P_{n-1}$ ，再给出凸多边形内部（含边界）一点  $Q$ 。现在要将这个凸多边形在地上无滑动地滚动一周，初始时  $P_0P_1$  边与地面接触，假设当前是  $P_iP_{(i+1) \bmod n}$  边与地面接触，那么滚动一下之后则是  $P_{(i+1) \bmod n}P_{(i+2) \bmod n}$  边与地面接触。不难发现，从初始状态滚动  $n$  下之后  $P_0P_1$  边再一次与地面接触，这时认为凸多边形已经滚动了一周。现在你需要求出凸多边形滚动一周之后点  $Q$  经过的轨迹长度。

### 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数。

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含一个整数  $n$ ，表示凸多边形的顶点数。

接下来  $n$  行，每行包含两个整数  $x_{P_i}, y_{P_i}$ ，按照逆时针的顺序给出凸多边形  $n$  个顶点  $P_1, P_2, \dots, P_n$  的坐标。

最后一行包含两个整数  $x_Q, y_Q$ ，表示点  $Q$  的坐标。

保证点  $Q$  在凸多边形内部（含边界）。

## 输出格式

对于每组测试数据，输出一行信息 "Case #x: y"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  表示凸多边形滚动一周之后点  $Q$  经过的轨迹长度，四舍五入精确到小数点后 3 位，数据保证答案的小数点后第 4 位不是 4 或 5。

## 题目 C. Caesar Cipher

在密码学中，恺撒密码是一种最简单且最广为人知的加密技术。它是一种替换加密的技术，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。

例如，当偏移量是 19 的时候，明密对照表如下：

明：A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

密：T U V W X Y Z A B C D E F G H I J K L M N O P Q R S

现在给出一段明文以及由这段明文加密得到的密文，你需要对另一段由相同加密方式进行加密的密文进行解密。

## 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数。

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含两个整数  $n, m$ 。

第二行和第三行分别包含一个长为  $n$  的只包含大写英文字母的字符串，分别表示一段明文以及由这段明文加密得到的密文；

第四行包含一个长为  $m$  的只包含大写英文字母的字符串，表示需要进行解密的密文。

## 输出格式

对于每组测试数据，输出一行信息 "Case #x: T"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $T$  表示密文解密得到的明文。

## 题目 D. Take Your Seat

朵哈要坐飞机去旅游，出发的时候有包括朵哈在内一共有  $n$  个人要上飞机。这班飞机有  $n$  个座位，第  $i$  个人的座位号是  $i$ ，朵哈的座位号就是 1。

现在  $n$  个人按照座位号是 1 到  $n$  的顺序上飞机。但朵哈忘了自己的座位，于是她就等概率随机选一个位置坐下了。而其余人都记得自己的座位。如果他们中的一个人上飞机后发现自己的位子被占了，则会在剩下的位置中等概率随机选一个坐下；如果没被占，则会直接坐在自己的位置上。

你需要计算最后一个上飞机的人坐到了自己位置上的概率。

朵哈返程的时候，包括她在内的  $m$  个人会按照一个随机的座位号顺序上飞机。

这班飞机有  $m$  个座位，朵哈的座位号还是 1。但是朵哈又一次忘记了自己的座位，且只有她忘记了自己的座位。现在所有人找座位的规则和出发时完全相同：朵哈或任何一个发现自己座位已经被占了的人等概率随机选一个没被占的座位坐下。

你需要计算最后一个上飞机的人坐到了自己位置上的概率。

## 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数。

接下来  $T$  行，每行描述一组测试数据，包含两个整数  $n$  和  $m$ ，分别表示出发时和返程时上飞机的人数。

## 输出格式

对于每组测试数据，输出一行信息 "**Case #x:**  $y$   $z$ "（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  表示出发时最后一个人坐到了自己位置上的概率， $z$  表示返程时最后一个人坐到了自己位置上的概率，四舍五入精确到小数点后 6 位，数据保证答案的小数点后第 7 位不是 4 或 5。

## 题目 E. 2-3-4 Tree

给一个空的 2-3-4 树，再给你一个  $n$  的全排列，按照这个全排列依次插入数字。

你需要输出最终树形态：按照先序遍历输出每个节点的信息，同一个节点的内的值按照从小到大的顺序输出。

（2-3-4 树的定义以及插入过程的详细描述，请参见原题。）

## 输入格式

第一行包含一个整数  $T$ ，表示有  $T$  组测试数据。

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含一个整数  $n$ ，表示排列的长度。

第二行包含  $n$  个整数  $p_1, p_2, \dots, p_n$ ，表示排列的  $n$  个数。

保证  $p_1, p_2, \dots, p_n$  是  $1, 2, \dots, n$  的一个排列。

## 输出格式

对于每组测试数据，先输出一行信息 "**Case #x:**"（不含引号），其中  $x$  表示这是第  $x$  组测试数据，接下来若干行，按照先序遍历输出每个节点的信息，每行包含若干个整数，按照从小到大的顺序输出这个节点中的值。

## 题目 F. Moving On

法尔杜丝和法图娜所在的国家有  $n$  个城市，编号 1 到  $n$ 。每一个城市都有一个危险系数。

现在有多次询问，每次询问给定两个城市  $u$  和  $v$ ，请找出一条从  $u$  到  $v$  的最短路径，使得途中经过的所有城市（不含起点和终点）的危险系数都不超过给定值  $w$ 。

## 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数。

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含两个整数  $n, q$ ，分别表示据点数和询问数。

第二行包含  $n$  个整数  $r_1, r_2, \dots, r_n$ ，表示每个据点的危险值。

接下来  $n$  行，每行包含  $n$  个整数，其中第  $i$  行第  $j$  列的元素是  $d_{i,j}$ ，表示从第  $i$  个据点直接移动到第  $j$  个据点需要的路程。

接下来  $q$  行，每行包含三个整数  $u_i, v_i, w_i$ ，表示要从第  $u_i$  个据点出发到第  $v_i$  个据点且所有途径的据点的危险值均不超过  $w_i$ 。

## 输出格式

对于每组测试数据，先输出一行信息 "Case #x:"（不含引号），其中  $x$  表示这是第  $x$  组测试数据，接下来  $q$  行，每行包含一个整数，表示这组询问的答案。

## 题目 G. Factories

比特大陆有  $n$  座城市以及  $n - 1$  条连接这些城市的双向道路，从任意一个城市出发沿着这些道路能到达任意其他城市，也就是构成了一棵  $n$  个点的树。

现在比特大陆要修建  $k$  座工厂，为了避免污染过大，每座工厂只能建在叶子节点上，并且每个叶子节点只能建一座工厂，但是工厂之间的运输效率也不能太低，因此要最小化任意两座工厂之间的距离之和，你需要计算出这个最小的距离和。

## 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数。

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含两个整数  $n, k$ ，表示比特大陆的城市数以及要修建的工厂数。

接下来  $n - 1$  行，每行包含三个整数  $u_i, v_i, w_i$ ，表示有一条连接  $u_i$  和  $v_i$  的长度为  $w_i$  的道路。

## 输出格式

对于每组测试数据，输出一行信息 "Case #x: y"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  表示最小的距离和。

## 题目 H. Fight Against Monsters

现在有  $n$  只怪兽，每只怪兽有一个体力值  $HP_i$  和一个攻击值  $ATK_i$ 。

英雄需要同时和这  $n$  只怪兽进行战斗。

在每一秒，首先英雄会被当前未被打倒的所有怪兽攻击，受到与这些怪兽的攻击值之和等量的伤害。

然后他要选择一只未被打倒的怪兽进行攻击。对同一只怪物进行的第  $i$  次攻击能对其造成  $i$  点伤害。

当怪兽的体力值  $\leq 0$  的时候就会倒下，当所有怪兽都被打倒时战斗立即结束。

英雄需要合理地进行攻击以使战斗过程中受到的伤害之和最小，请你求出这个伤害总量。

### 输入格式

第一行包含一个整数  $T$ ，表示有  $T$  组测试数据。

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含一个整数  $n$ ，表示怪兽的数量。

接下来  $n$  行，每行包含两个整数  $HP_i, ATK_i$ ，表示每只怪兽的体力值和攻击值。

### 输出格式

对于每组测试数据，输出一行信息 "Case #x: y"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  表示战斗过程中受到的伤害之和的最小值。

## 题目 I. Bubble Sort

给定正整数  $n, k$  和素数  $q$ ，依兹迪哈尔想知道有多少个由正整数 1 到  $n$  组成的排列  $a$  在冒泡排序  $k$  趟之后能得到一个“几乎被正确排序”的序列。（冒泡排序的伪代码请参见原题。）

一个序列被称作“几乎被正确排序”是指它的最长上升子序列长度至少为  $n - 1$ 。这个数值可能很大，你只需要给出这个数值对  $q$  取模后的值。

### 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数。

接下来  $T$  行，每行描述一组测试数据，包含三个整数依次为  $n, k$  和  $q$ 。

### 输出格式

对于每组测试数据，输出一行信息 "Case #x: y"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  表示这组测试数据的答案对  $q$  取模的值。

## 题目 J. Nested Triangles

给定平面上 2 个点  $P$  和  $Q$ ，我们称它们为支点。再给定若干其他的点，记为  $A_1, A_2, \dots, A_{n-1}$  和

$A_n$ 。保证它们中每一个都不与  $P, Q$  共线。

现在你需要找出以  $P, Q$  为支点的最大嵌套三角形方案。一个嵌套三角形方案是指选择一系列点  $A_{v_1}, A_{v_2}, \dots, A_{v_k}$  满足对于任意的  $i \geq 2$  来说，点  $A_{v_i}$  总是落在三角形  $PQA_{v_{i-1}}$  内（不允许落在边界上）。

如果有多组最大嵌套三角形方案，请输出字典序最小的一个。也就是说输出  $v_1$  最小的方案；如果仍然有多个方案，输出  $v_2$  最小的方案；如果依然有多个方案，输出  $v_3$  最小的方案，并以此类推下去。

## 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数，

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含四个整数  $x_P, y_P, x_Q, y_Q$  给出了支点  $P$  和  $Q$  的坐标。

第二行包含一个整数  $n$ ，表示给出的其它的点的个数。之后  $n$  行中的每一行包含两个整数，描述了一个给定的点的坐标。

保证所有给定的点两两不相同。

## 输出格式

对于每组测试数据，首先输出一行信息 "Case #x: y"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  表示最大嵌套三角形方案中选择的点的个数。之后  $y$  行中的第  $i$  行含有一个整数  $v_i$  表示选中的第  $i$  个点的下标。

## 题目 K. Vertex Covers

给一个无自环无重边的无向图  $G$ ，点有点权，计算所有点覆盖集的点权乘积（空集的点权乘积为 1）的和，对给定的一个素数  $q$  取模后输出。

## 输入格式

第一行包含一个整数  $T$ ，表示测试数据的组数，

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含三个整数  $n, m, q$ ，表示图  $G$  的点数和边数以及给定的素数模数。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$ ，表示每个点的权值。

接下来  $m$  行，每行包含两个整数  $u_i, v_i$ ，表示  $u_i$  号点和  $v_i$  号点之间有一条边。

## 输出格式

对于每组测试数据，输出一行信息 "Case #x: y"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  表示所有点覆盖集的点权乘积之和对  $q$  取模后的值。

## 题目 L. Continuous Intervals

给一个长为  $n$  的整数序列，你需要统计有多少个连续区间。

一个区间被称为“连续区间”是指：如果将区间内的数按照从小到大排序，则排序后任意相邻两数之差不超过 1。

### 输入格式

第一行包含一个整数  $T$ ，表示有  $T$  组测试数据。

接下来依次描述  $T$  组测试数据。对于每组测试数据：

第一行包含一个整数  $n$ ，表示序列的长度。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$ ，表示给定的整数序列。

### 输出格式

对于每组测试数据，输出一行信息 "Case #x: y"（不含引号），其中  $x$  表示这是第  $x$  组测试数据， $y$  表示连续区间的个数。

## 题目 M. Acyclic Orientation

给一个两部分点数分别为  $n$  和  $m$  的完全二分图（一共有  $n \cdot m$  条无向边），现在给每条边确定一个方向，你需要计算有多少种方案使得定向后的图是有向无环图。

对于无向图  $G$  来说，定义它的染色多项式为  $\chi_G(k)$ ，这是一个关于  $k$  的  $|V(G)|$  次多项式。其中当  $k$  为整数的时候  $\chi_G(k)$  表示用  $k$  种颜色对图  $G$  染色使得每条边的两个端点颜色不同的方案数。

染色多项式有诸多应用，比如说无向图的无环定向问题。

对于一个无向图  $G$  来说，它的一个无环定向方案是给每一条边都确定了一个方向，使得定向后的图是有向无环图。图  $G$  的无环定向方案数正是  $|\chi_G(-1)|$ ：需要说明的是，虽然用  $-1$  种颜色对图染色在组合意义下是说不通的，但是由于  $\chi_G(k)$  是个多项式，代入  $k = -1$  进行求值在代数意义下是合理的。

现在给你一个两部分点数分别为  $n$  和  $m$  的完全二分图（一共有  $n \cdot m$  条无向边），请求出来它的无环定向方案数。

### 输入格式

第一行包含一个正整数  $T$ ，表示测试数据的组数。

接下来  $T$  行，每行描述一组测试数据，包含三个正整数  $n, m$  和  $q$ ，依次表示完全二分图两部分的点数以及给定的素数模数。

### 输出格式

对于每组测试数据，输出一行信息 Case #x: y，其中  $x$  表示这是第  $x$  组测试数据， $y$  表示这组测试数据的答案对  $q$  取模的值。