

Problem A: 村落中的长者

时间限制: 1000MS 内存限制: 65536KB

题意描述:

在一个村落里, 有 n 个人, 每个人的资历都*互不相同*, 由一个非负整数 a_i 表示, a_i 越大表示这个人资历越高。现在, 由于某些原因, 村落中的人们组成了小团体, 小团体的人是村落中所有人的一个非空子集 (易知共有 $2^n - 1$ 个可能的非空子集)。在某个确定的小团体中, 资历最高的人称为“长者”, 长者的资历值称为这个小团体的 E 值。试问, *所有可能* 的小团体的 E 值之和是多少?

输入:

首先输入一个正整数 T , 表示测试数据的组数。接下来每组测试数据, 第一行输入一个正整数 n , 表示村落中人的个数。第二行输入 n 个非负整数, 表示这 n 个人的资历值。

输出:

对于每组测试数据, 输出一行 “Case #x: answer”, 其中 x 为测试数据的编号, answer 为相应的答案。

数据范围:

$$1 \leq T \leq 1000$$

$$1 \leq n \leq 20$$

$$0 \leq a_i \leq 100$$

样例输入：

2

1

66

3

1 2 3

样例输出：

Case #1: 66

Case #2: 17

样例解释：

对于第一组样例，村落中只有一位村民，可能构成的非空子集只有 {66} 这一个，故而 E 值之和为 66；对于第二组样例，村落中有三位村民，可能构成的非空子集有 {1}，{2}，{3}，{1,2}，{1,3}，{2,3}，{1,2,3}，E 值分别为 1，2，3，2，3，3，3，所以 E 值之和为 17。

Problem B: とある不科学の方程式

时间限制: 1000MS 内存限制: 65536KB

题意描述:

对于下面的方程: $x_1 \text{ xor } x_2 \text{ xor } \dots \text{ xor } x_n = k$, 其中 xor 表示异或运算, 且其中所有的变量均满足 $0 \leq x_i \leq m_i, 1 \leq i \leq n$ 。现在给出 m_i, n, k , 试问该方程共有多少组不同的解?

输入:

首先输入一个正整数 T , 表示测试数据的组数。接下来 T 组数据, 每组的第一行为正整数 n 和 k , 第二行为 n 个正整数表示 m_i 。

输出:

对于每组测试数据, 输出一行 “Case #x: answer”, 其中 x 为测试数据的编号, answer 为对应的答案。由于 answer 可能非常大, 故而请输出 answer 对 1000000003 ($10^9 + 3$) 取模后的结果。

数据范围:

$$1 \leq T \leq 100$$

$$1 \leq n \leq 50$$

$$0 \leq k \leq 2^{31} - 1$$

$$0 \leq m_i \leq 2^{31} - 1$$

样例输入：

2

11 2047

1024 512 256 128 64 32 16 8 4 2 1

10 2047

1024 512 256 128 64 32 16 8 4 2

样例输出：

Case #1: 1

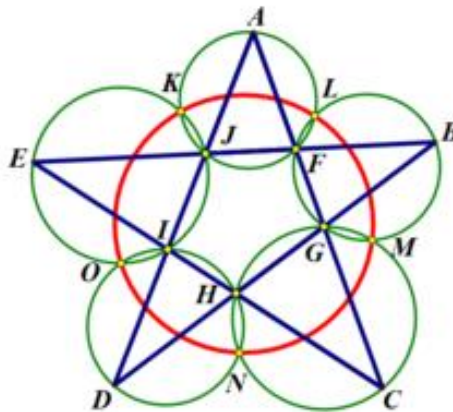
Case #2: 0

Problem C: 五点共圆

时间限制: 1000MS 内存限制: 65536KB

题意描述:

一位合格的魔法师一定见过下面这个题目: 在任意五角星 $AJEIDHCGBF$ 中, $\triangle AFJ$ 、 $\triangle JEI$ 、 $\triangle IDH$ 、 $\triangle HCG$ 和 $\triangle GBF$ 各自的外接圆顺次相交的交点分别是 K 、 O 、 N 、 M 、 L 。求证 K 、 O 、 N 、 M 、 L 五点共圆。我们自然不能要求新人们解决一个姿势水平如此之高的几何题, 于是请考虑下面这样一个简化版的问题: 给出平面上五个点 A 、 B 、 C 、 D 、 E , 判断这五个点是否共圆。



输入:

首先输入正整数 T , 表示测试数据的组数。每组数据包含十个整数, 依次表示五个点的横纵坐标, 即 $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, x_5, y_5$ 。数据保证这五个点互不相同。

输出:

对于每组测试数据, 输出一行 “Case #x: answer”, 其中 x 是测试数

据的编号。若给出的五点确实共圆，则 answer 为 Yes，否则为 No。

数据范围：

$$1 \leq T \leq 100$$

$$-1000 \leq x_i, y_i \leq 1000$$

样例输入：

2

1 0 1 1 0 1 -1 0 0 -1

2 1 1 2 2 -1 1 -2 -1 2

样例输出：

Case #1: No

Case #2: Yes

Problem D: 哆啦 A 梦的口袋

时间限制: 3000MS 内存限制: 65536KB

题意描述:

这天, 大雄带了一大群小伙伴回家, 想要一起在秘密基地中玩玩具。这可难坏了哆啦 A 梦, 因为每个秘密基地的玩具是有限的, 要保证每个人都有玩具玩就只能开启 N 个秘密基地 (N 个基地玩具数量保证足够每人一个玩具), 已知每个基地玩具的数量 (非负整数), 初始时, 哆啦 A 梦随机把小朋友们分到了 N 个基地中, 每个基地有 a_i 个玩具, 分到了 b_i 个小朋友, 但是 a_i 和 b_i 谁大谁小不一定 (有的基地有人没有玩具, 有的基地有空闲玩具)。为了保证大家都有玩具, 哆啦 A 梦在几个基地之间搭了 M 个传送门 (每个传送门是**双向的**, 没有玩具的小朋友可以通过传送门去另一个有空闲玩具的基地玩, 且不用回来), 但是每个传送门最多传送 c_i 个人, 且有个问题, 就是在传送了两个人后, **从第三个人开始**就可能出现故障, 故障概率为 p_i (每个传送门的 p_i 不相同)。问, 在使得每个人都有玩具的前提下, 如何使**所有传送门都出现故障的概率**最小。

输入:

首先输入一个正整数 T , 表示测试数据的组数。接下来 T 组数据, 每组第一行输入两个整数 N, M 。接下来的 N 行, 每行两个整数, 第一个为该基地 i 初始时随机到了 b_i 个小朋友, 第二个为基地 i 拥有的玩具数量 a_i 。接下来 M 行, 每行三个整数一个浮点数, 前两个整数为一个

传送门的两端基地的编号 u, v , 第三个整数为该传送门最多运 c_i 个人, 浮点数为第三个人开始可能出现故障的概率 p_i 。

输出:

对于每组测试数据, 输出一行 “Case #x: answer”, 其中 x 为测试数据的编号, answer 为在保证每个人有玩具的前提下所有传送门都出现故障的最小概率 (保留 *小数点后两位*)。

数据范围:

$$1 \leq T \leq 10$$

$$1 \leq N \leq 100$$

$$1 \leq M \leq 5000$$

$$0 \leq a_i, b_i \leq 200$$

$$1 \leq c_i \leq 100$$

$$0 \leq p_i < 1$$

样例输入:

1

4 4

2 0

0 3

3 0

0 3

1 2 5 0.5

3 2 5 0.5

1 4 5 0.5

3 4 5 0.5

样例输出：

Case #1: 0.00

Problem E: 灵魂链接

时间限制: 1500MS 内存限制: 65536KB

题意描述:

包子和老司机这一天在进行魔兽争霸 3 的对战, 老司机使用兽族而包子使用不死族。老司机擅长使用白牛的灵魂链接和巫医的团补技能, 灵魂链接就是将两个没有链接的单位链接起来共同承受伤害, 团补技能就是对一个或者多个单位一定恢复生命值。现在老司机集齐了 N 个士兵准备进攻包子的基地, 请你帮帮包子对抗邪恶的老司机!

输入:

首先输入一个正整数 T , 表示测试数据的组数。每一组测试数据, 首先输入一个 n 和 m , 表示老司机派了 n 个士兵 (编号为 $1 \sim n$) 进攻包子的基地以及 m 个操作, 最开始 n 个士兵都没有被链接。接下来输入 n 个数 a_i , 表示 n 个士兵的初始生命值。然后输入 m 个操作, 输入的操作可能是以下几种:

- (1) link x y : 老司机使用白牛链接了第 x 个士兵以及第 y 个士兵;
- (2) add1 x v : 老司机对第 x 个士兵补充了 v 点生命值;
- (3) add2 x v : 老司机对第 x 个士兵及其链接的所有单位补充了 v 点生命值;
- (4) add3 v : 老司机对所有士兵补充了 v 点生命值;
- (5) find1 x : 输出第 x 个士兵目前的生命值;
- (6) find2 x : 输出第 x 个士兵所在的链接团体中生命值最少的士兵

的生命值；

(7) find3: 输出老司机目前军队中生命值最少的士兵生命值。

输出：

对于每组测试样例，首先输出一行“Case #x:”，其中 x 为测试数据的编号。接下来，对于该组样例中的每一个 find 类型的操作，输出相应的结果。

数据范围：

$$1 \leq T \leq 10$$

$$0 \leq n, m \leq 300000$$

$$0 < a_i, v \leq 1000$$

样例输入：

1

3 4

1 1 1

find1 1

link 1 2

add2 2 1

find3

样例输出:

Case #1:

1

1

Problem F: 上课小动作

时间限制：500MS 内存限制：65536KB

题意描述：

上课困了怎么办，趴下睡觉？J老师说：困的时候把题目拿出来做做，就有精神了。大师W觉得课程内容太简单，一不小心就犯困了。他记起老师的话，便想办法出了一道题给自己做：他拿出他新买的 n 支颜色不同的彩色笔，给这些笔编号为 $1 \sim n$ 。在一条宽度为 1cm ，长度无限的纸带上（大师W的书包里总是有些神奇的东西），包含很多 1×1 的方格，将它们按照 $1、2、3 \cdots$ 编号。现在从格子 1 开始，大师W开始用彩色笔依次给方格涂色。如果某个方格用的是编号为 x 的笔，那么下一个格子就必须用 y 号笔涂色。其中 $y = \begin{cases} x + 1, & x \neq n \\ 1, & x = n \end{cases}$ 。这样大师W就能在课上消磨时间了。不过大师的每支笔墨水都有限，只能涂有限数量个格子。大师W显然很快就算出来最多能涂多少个格子，又陷入了无聊状态。现在请问聪明的你解决这个问题：大师W按照这种方式，最多能涂多少个格子呢？

输入：

首先输入正整数 T ，表示测试数据的组数。每组数据首先输入 n ，表示大师W拥有多少支彩色笔。下面一行中给出 n 个数字 a_i ，第 i 个数字表示编号为 i 的笔最多能涂的方格数。

输出：

对于每组测试数据，输出一行“Case #x: answer”，其中 x 是测试数据的编号。answer 表示大师 W 最多能涂的格子数。

数据范围：

$$1 \leq T \leq 100$$

$$1 \leq n \leq 100$$

$$0 \leq a_i \leq 10^6$$

样例输入：

3

5

2 4 2 3 3

3

5 5 5

6

10 10 10 1 10 10

样例输出：

Case #1: 12

Case #2: 15

Case #3: 11

Problem G: 艾神的难题

时间限制: 3000MS 内存限制: 65536KB

题意描述:

艾神不仅智商超群而且极其幽默，情商极高，深受几个妹子的喜爱 2333。这天，艾神突发奇想，给了妹子一个 $1 \sim n$ 的某种排列（例如： $n=11$ ，给定的全排列为 1 3 2 5 4 6 9 8 7 10 11）。他允许一种操作，且可以做无数次：她们每次可以选出任意三个数字，将三个数字随意排列后再放回到序列中原来的位置。例如对于上面的序列，选择 1 2 4，并排序为 1 4 2，然后放回去，放回去后的序列为 1 3 4 5 2 6 9 8 7 10 11。其中选择 1 2 4 的概率为 $1/C_{11}^3$ ，将他们排序为 1 4 2 的概率为 $1/6$ （三个数字排列有 6 种）。艾神想要最后得到从 1 到 n 的序列（对于 $n=11$ 时即 1 2 3 4 5 6 7 8 9 10 11）。试问最后达到艾神的要求的平均期望操作次数是多少。（对于每一种初始排列的期望操作次数是指 $\sum \text{操作次数} \times \text{该次数的概率}$ ，且平均期望指 $\text{每种初始排列下期望操作次数的和} / n!$ ）

输入:

首先输入正整数 T ，表示测试数据的组数。每组数据输入一个正整数 n 。

输出:

对于每组测试数据，输出一行 “Case #x: answer”，其中 x 是测试数

据的编号。answer 表示平均期望值（输出*从左到右四位数字*，例如 12345.234 输出 1234，12.356 输出 1235）。

数据范围：

$$1 \leq T \leq 9$$

$$3 \leq n \leq 11$$

样例输入：

2

3

4

样例输出：

Case #1: 5000

Case #2: 2750

Problem H: 看似复杂的方程

时间限制: 1000MS 内存限制: 65536KB

题意描述:

有许多看似复杂的方程实际上有着非常简单的解。考虑下面的方程:

$x! + y! = x^y$, 其中 x 和 y 均为正整数。这显然是一个不定方程, 如果

告诉你 y 的值, 你是否可以找到一个 x 使得方程成立呢?

输入:

首先输入一个正整数 T 表示测试数据的组数。接下来 T 组数据, 每组都包含一个正整数 y 。

输出:

对于每组测试数据, 输出一行 “Case # x : answer”, 其中 x 为测试数据的编号, 如果对于给出的 y 存在正整数 x 使得上述方程成立, 则 answer 为 Yes, 否则为 No。

数据范围:

$$1 \leq T \leq 10000$$

$$1 \leq y \leq 10000$$

样例输入:

2

1

2

样例输出：

Case #1: No

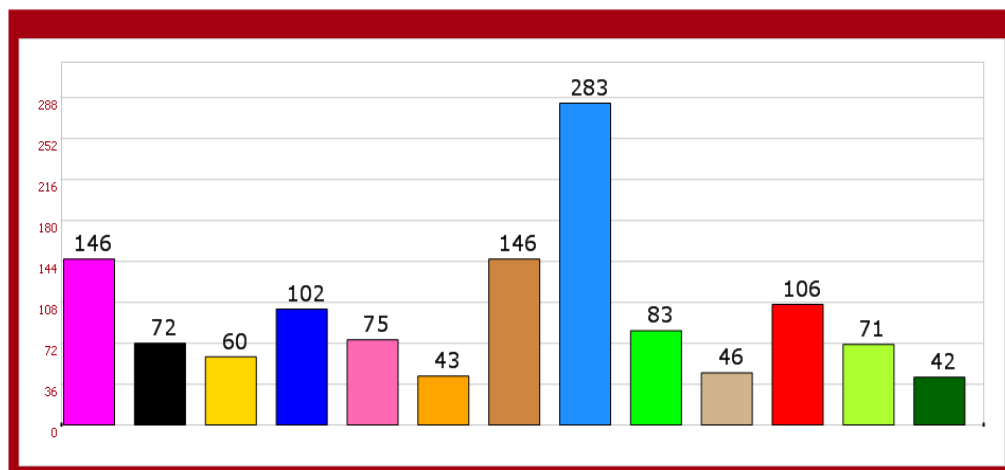
Case #2: Yes

Problem I: 柱状图排序

时间限制：1000MS 内存限制：65536KB

题意描述：

柱状图是一种使用频率很高的统计图表，而一种常见的操作就是对柱状图中的各个项目的大小进行排序。现在请你实现这个功能。



输入：

首先输入一个正整数 T ，表示测试数据的组数。接下来 T 组数据，每组数据由 m 行**长度相等**的字符串 $s_1 \sim s_m$ 构成，表示一个柱状图。这些字符串的最后一行一定形如 “ABC...”，每个大写英文字母表示柱状图的一个项目名；之前的每一行都由字符 “#” 和字符 “\$” 构成，每一个 “#” 表示一个单位高度的矩形，“\$” 表示空格，某一项目名上的 “#” 越多则该项目的值就越大。输入数据保证是一个合法的柱状图，即每一个项目上都最多只有一段连续的 “#”，形如一个长长的柱。

输出：

对于每组数据，首先输出一行“Case #x:”，其中 x 为测试数据的编号。接下来输出一个正整数 n，占一行，表示这个柱状图中项目的个数。

然后按照值从大到小的顺序输出 n 行，每行形如“X num”，X 是一个大写英文字母，表示一个项目名，num 是这个项目的值。如果两个项目的值相同，则按*项目名在字母表中的顺序*输出。

数据范围：

$$1 \leq T \leq 100$$

$$1 \leq m \leq 100$$

$$1 \leq \text{length}(s_i) \leq 26 \text{ 且 } \text{length}(s_i) = \text{length}(s_j)$$

样例输入：

2

\$#\$

\$##

###

ABC

\$\$\$\$#

##\$\$#

##\$##

#####

#####

ABCDE

样例输出：

Case #1:

3

B 3

C 2

A 1

Case #2:

5

E 5

A 4

B 4

D 3

C 2

样例解释：

在第一组样例中，A 的值为 1，B 的值为 3，C 的值为 2；在第二组样例中，A 和 B 的值均为 4，按照题意 A 排在 B 的前面。

Problem J: 疯狂的游戏

时间限制: 1000MS 内存限制: 65536KB

题意描述:

与擅长数学的 Dr. Shen 玩游戏向来是一件疯狂的事情。很不幸, 你被指派来与 Dr. Shen 玩一个疯狂的游戏。桌面上摆有 n 张纸牌, 依次写有数字 $0, 1, \dots, n-1$ 。Dr. Shen 使用他的魔法, 偷偷将每张牌上的数字都增大了 m (然而 **你并不知道 m 是多少**)。现在 Dr. Shen 提出了一个无理的要求: 他希望你找到一个最小的正整数 t , 使得他从这 n 张牌中任取 t 张, 其中都存在 3 张牌, 满足这 3 张牌上的数字两两互素 (互素是指最大公约数为 1)。

输入:

首先输入一个正整数 T , 表示测试数据的组数。接下来 T 组数据, 每组都包含一个正整数 n 。

输出:

对于每组测试数据, 输出一行 “Case #x: answer”, 其中 x 为测试数据的编号, answer 为所求的 t 的最小值。

数据范围:

$$1 \leq T \leq 1000$$

$$4 \leq n \leq 10000$$

$$1 \leq m$$

样例输入：

3

4

5

6

样例输出：

Case #1: 4

Case #2: 5

Case #3: 5

Problem K: 雪风酱是不会沉的!

时间限制: 3000MS 内存限制: 65536KB

题意描述:

提督又派雪风出击了。出击海域有 N 个点, 每个点拥有一个攻略难度值 D , 雪风希望将这 N 个点任意分为 M 组分别攻略 ($M < N$), 每组包含至少一个点, 每个点只能被一个组包含。每组中难度**最大和最小值差的平方**为该组的难度值。现在雪风想知道, 出击海域的所有组的难度值之和最小是多少。

输入:

首先输入正整数 T , 表示测试数据的组数。每组数据第一行输入 N 和 M , 表示出击海域的点数和雪风想要分组的数量。随后的一行包含 N 个数, 表示每个点的攻略难度。

输出:

对于每组测试数据, 输出一行 “Case #x: answer”, 其中 x 是测试数据的编号, answer 表示海域最小的难度值。

数据范围:

$$1 \leq T \leq 20$$

$$1 \leq N \leq 10000$$

$$1 \leq M \leq 5000 \text{ 且 } M < N$$

$$0 \leq D_i \leq 9000$$

样例输入：

2

3 2

1 2 4

4 2

4 7 10 1

样例输出：

Case #1: 1

Case #2: 18

Problem L: 魔法！真正的魔法！

时间限制：3000MS 内存限制：65536KB

题意描述：

某日，魔法师教会的 n 名魔法师（包含一个大魔导师）进行一次重要的大规模施法，用于抵抗敌对阵营。在施法现场，有 m 个魔法阵排成一行，编号为 $1 \sim m$ 。然而，由于敌对阵营的情报网十分强大，他们提前摧毁了其中某些魔法阵，使得这些魔法阵不再生效。施法开始时，每位魔法师都需要站到一个有效的魔法阵里，且每个魔法阵里最多只能站一个人。另外，这个教会认为“素数”是一种不吉利的数，故而**除了大魔导师以外**的人都不愿意站在编号为素数的魔法阵里。为了增强魔法的力量，应当使所有魔法师与大魔导师之间的距离的最大值最小（两个魔法师之间的距离定义为两人所站的魔法阵编号差的绝对值）。现在，教会希望你来帮他们算一下这个最小值。如果这场施法无法进行，你也需要告诉教会这个遗憾的事实。

输入：

首先输入一个正整数 T ，表示测试数据的组数。接下来 T 组数据，每组数据的第一行是两个正整数 n, m ，分别表示包含大魔导师在内的魔法师的人数以及魔法阵的个数；第二行是 m 个整数 $a_1 \sim a_m$ ，若 $a_i = 1$ 则表示编号为 i 的魔法阵已经被破坏了， $a_i = 0$ 表示编号为 i 的魔法阵尚未被破坏。

输出：

对于每组测试数据，输出一行 “Case #x: answer”，其中 x 为测试数据的编号，answer 为该组样例中最小的距离最大值，若这场施法无法进行，answer 为 “So Sad”。

数据范围：

$$1 \leq T \leq 1000$$

$$2 \leq n \leq 10000$$

$$1 \leq m \leq 10000$$

$$a_i \in \{0,1\}$$

样例输入：

3

2 3

0 1 0

3 3

0 0 0

3 5

0 0 1 0 0

样例输出：

Case #1: 2

Case #2: So Sad

Case #3: 2

样例解释：

对于第一组样例，编号为 1 和 3 的魔法阵可以使用，1 不是素数，故而可以让大魔导师站在 3 号魔法阵，另一个魔法师站在 1 号魔法阵，距离为 2；对于第二组样例，虽然三个魔法阵都可以使用，但是 2 和 3 都是素数，故而不可能使三个魔法师各站到一个魔法阵里；对于第三组样例，可以使用的魔法阵为 1, 2, 4, 5，故而两个魔法师必须站在 1 号和 4 号，如果大魔导师站在 2 号，那么他和另外两个魔法师的距离分别为 1 和 2，最大值为 2，如果大魔导师站在 5 号，那么他和另外两个魔法师的距离分别为 4 和 1，最大值为 4，从而距离最大值的最小值为 2。