

2021 年 **大连大学** ACM 校内选拔赛
(正式赛)



主办单位：大连大学创新创业学院

承办单位：大连大学 ACM 程序设计工作室

2021 年 4 月 25 日

Problem A. Made in Abyss

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



“……阿比斯是一个巨大的洞穴，约 1900 年前于南海佩奥鲁斯卡的孤岛上被发现，直径 1000m，推测其深度在 20000m 以上。在深渊中存在着许多未知文明的遗迹和无数神奇的遗物，驱使着探险家们前赴后继地进入阿比斯。……”

你作为贝尔切洛的一名持铃生，在课堂上打瞌睡是常有的事。但明天有一场随堂小测验：试卷上将会有多个数字，每个数字都代表阿比斯的深度，对于每个深度，你需要填写该深度的区域名称。

虽然你的记性和判断力都不太好，但你十分擅长编程，所以你可以写一个程序帮助你通过考试。

Input

多组测试样例。输入的第一行为一个正整数 $T(1 \leq T \leq 10^5)$ ，代表测试样例组数。

接下来 T 行，每一行都有一个非负整数 $d(0 \leq d \leq 10^5)$ ，代表阿比斯的深度。

Output

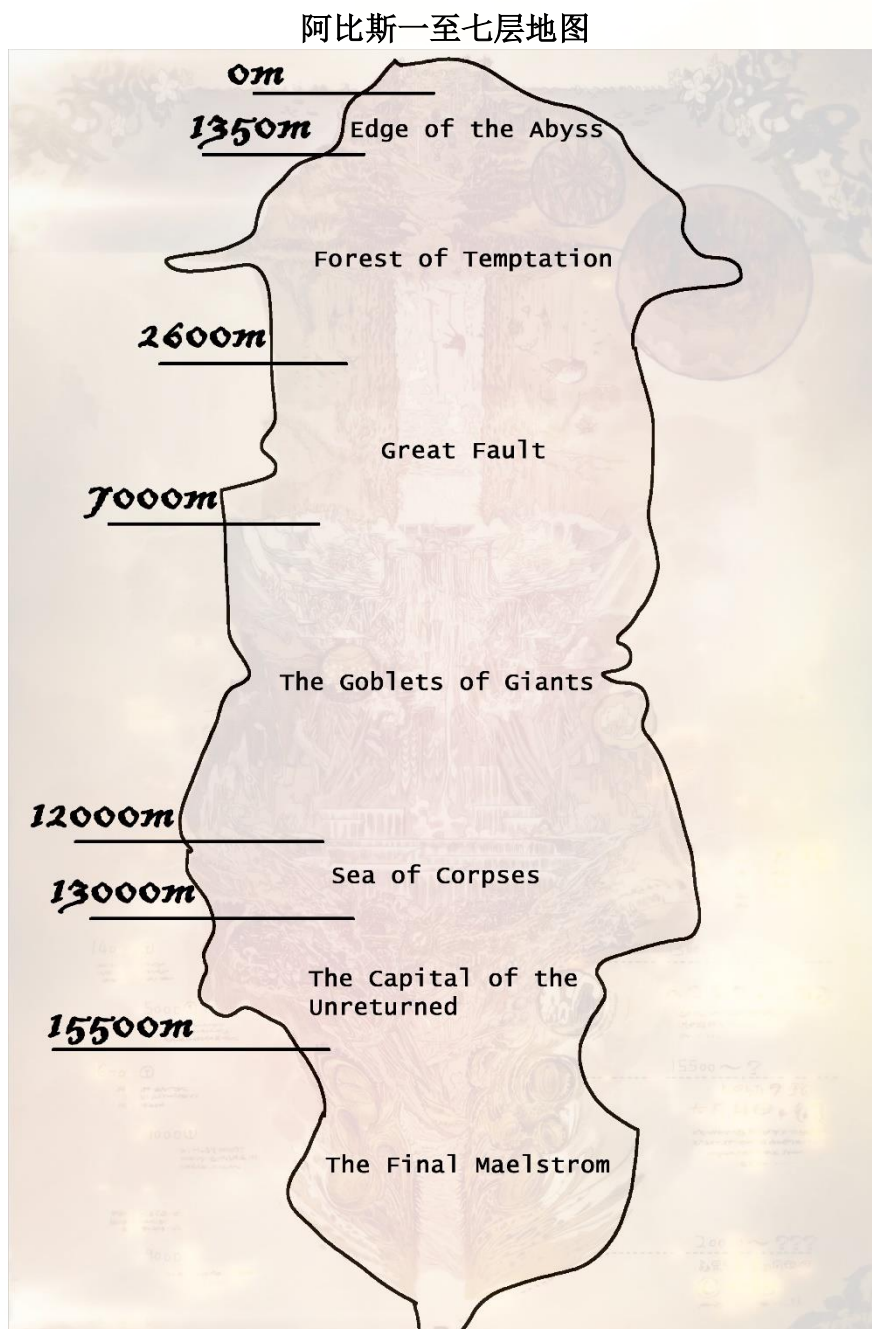
对于每一个深度，对应区域名称如下（输出不带双引号，注意大小写和换行）：

- $d \in [0, 1349]$ ，输出 “Edge of the Abyss”。
- $d \in [1350, 2599]$ ，输出 “Forest of Temptation”。
- $d \in [2600, 6999]$ ，输出 “Great Fault”。
- $d \in [7000, 11999]$ ，输出 “The Goblets of Giants”。
- $d \in [12000, 12999]$ ，输出 “Sea of Corpses”。
- $d \in [13000, 15499]$ ，输出 “The Capital of the Unreturned”。
- $d \in [15500, +\infty]$ ，输出 “The Final Maelstrom”。

Examples

standard input	standard output
3 16234 0 2354	The Final Maelstrom Edge of the Abyss Forest of Temptation

Explanations



Problem B. Best Match

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



给你一个长度为 n 的排列 p 和一个长度为 m 的数组 a ($a_i \in [1, n]$)。对于数组 a 的某对下标 $[l, r]$, 若能在 l 位置至 r 位置之间找到一个子数组与排列 p 的一个循环移位相同, 则称数组 a 在 $[l, r]$ 上和排列 p “完美匹配”。

长度为 n 的排列: 在一个长度为 n 的数组中, 数字 $[1, n]$ 都出现了恰好一次, 比如 $(3, 1, 2)$ 、 $(6, 3, 1, 4, 5, 2)$ 都是排列, 但 $(1, 3, 3)$ 、 $(6, 7, 3, 2, 1, 5)$ 都不是排列。

子数组: 在一个数组中, 任意连续的数字构成的子部分称为子数组。例如对于数组 $(1, 2, 3)$, 那么它有 (1) 、 (2) 、 (3) 、 $(1, 2)$ 、 $(2, 3)$ 、 $(1, 2, 3)$ 共 6 个子数组, 而 $(1, 3)$ 、 $(3, 2)$ 则不是它的子数组。

排列的循环移位: 对于排列 (p_1, p_2, \dots, p_n) , $(p_i, p_{i+1}, \dots, p_n, p_1, p_2, \dots, p_{i-1})$ ($i \in [1, n]$) 是原排列的一个循环移位。如对于排列 $(2, 1, 3)$, 那么它有 $(1, 3, 2)$ 、 $(3, 2, 1)$ 、 $(2, 1, 3)$ 共 3 个循环移位, 而 $(1, 2, 3)$ 、 $(3, 1, 2)$ 都不是它的循环移位。

现在有 q 次询问, 每个询问包含一对下标 $[l, r]$, 你需要回答 a 在 $[l, r]$ 上与排列 p 是否“完美匹配”。

Input

第一行为正整数 n, m, q ($1 \leq n, m, q \leq 2 \times 10^5$), 含义见题面。

第二行为排列 p , 第三行为数组 a ($a_i, p_i \in [1, n]$)。

接下来有 q 个询问, 第 i 个询问包含一对数组 a 的下标 l_i, r_i ($1 \leq l_i \leq r_i \leq m$), 注意询问是闭区间。

Output

对于每一个询问, 若数组 a 在 $[l, r]$ 上与排列 p “完美匹配”, 输出 YES, 否则输出 NO。

Examples

standard input	standard output
3 6 3	NO
2 1 3	YES
1 1 1 3 2 3	NO
1 4	
3 6	
3 4	

Explanations

第一个询问: [1,4]

1 1 1 3 2 3

显然黄色记号笔标注区间内, 不能找到排列 (2,1,3) 的循环移位, 因此数组 a 在 [1,4] 上和排列 p 不能完美匹配, 故输出 NO。

第二个询问: [3,6]

1 1 1 3 2 3

对于排列 (2,1,3) 的一个循环移位 (1,3,2), 显然黄色记号笔标注区间内存在该循环移位, 因此数组 a 在 [3,6] 上和排列 p 能完美匹配, 故输出 YES。

第三个询问: [3,4]

1 1 1 3 2 3

区间内显然不可能存在排列 (2,1,3) 的循环移位, 因为区间长度为 2, 但排列长度为 3, 故输出 NO。

Problem C. 5 Centimeters Per Second

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



远野贵树站在一排樱花树前。

现在正值樱花飘落的季节，花瓣下落的速度为 5cm/s 。贵树现在站在第一棵樱花树下，他可以花费 1 秒种到相邻的下一棵樱花树下。现在已知共有 n 棵樱花树，每棵樱花树都有 1 片花瓣即将掉落，给出各花瓣的高度，假设所有花瓣在一开始同一时间垂直向下飘落，他最多可以收集到多少片花瓣？

就如爱情一般浪漫，贵树可以站在原地等待樱花飘落；

就如爱情一般脆弱，当花瓣触地的瞬间，若贵树已到达花瓣所在的樱花树下，则视为他接到了花瓣，否则他就永远错失了这片花瓣；

就如爱情一般易逝，贵树只能朝着最后一棵樱花树进发而无法回头。

Input

第一行为正整数 n ($1 \leq n \leq 2 \times 10^5$)，代表花瓣个数。

第二行包含 n 个正整数，第 i 个正整数 h_i ($1 \leq h_i \leq 10^9, 5|h_i$) cm 代表第 i 棵樱花树上花瓣的高度。

注明：| 为整除符号， $a|b$ 等价于 $b\%a = 0$ 。

Output

输出远野贵树最多可收集到的花瓣数量。

Examples

standard input	standard output
8 15 20 10 30 35 45 10 40	5

Explanations

可以证明如下过程可以让贵树收集到最大 5 片花瓣：

1. 站在第一棵树下等待，在第 3s 结束的时候拿到第一片花瓣：

第三秒结束时：一片



2. 在第 4s 结束的时候，走到第二棵树下，正好接到第二片花瓣。

第四秒结束时：两片



3. 一直向前走，在第 6s 结束的时候走到第四棵树下接到第三片花瓣。（第三棵树的花瓣在他走到时已经落地，所以他无法接到）

第五秒结束时：两片

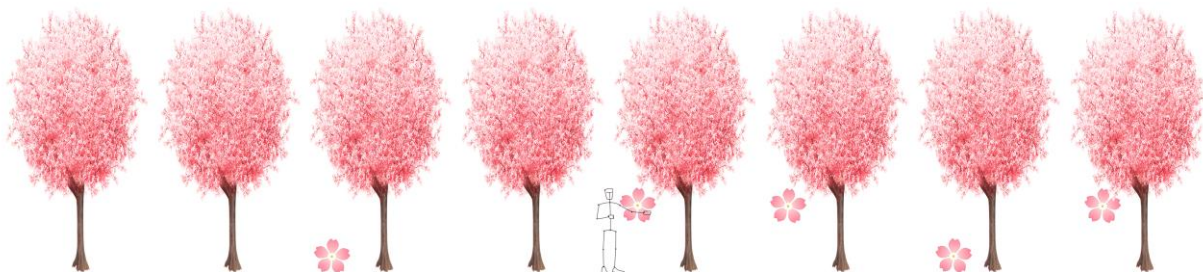


第六秒结束时：三片



4. 在第 7s 结束的时候，走到第五棵树下，正好接到第四片花瓣。

第七秒结束时：四片



5. 在第 8s 结束的时候，走到第六棵树下，驻足等待，在第 9s 结束时接到第五片花瓣。（此时第七棵和第八棵树上的花瓣已经落地，故贵树不必继续向前）。

第九秒结束时：五片



This page is intentionally left blank.
此页有意留为空白。

Problem D. No Netherite No PG

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes



PG 站在一个堡垒遗迹前。

这个堡垒遗迹内共有 n 个箱子可供 PG 进行开启，每个箱子含有 a_i 单位的下界合金资源，PG 每打开一个箱子，就可以将其中的下界合金资源全部拿完。这些箱子所在地间通过 m 条有向路径连接起来且不成环，即整个堡垒遗迹构成一张有向无环图。

每条路径上都有下界猪灵或疣猪兽把守，准确的说每条路径上有 w 单位的敌对生物，想要通过路径，就需要打败路径上的所有敌对生物。PG 审视了一下自身的装备，他总共最多只能解决 W 单位的敌对生物了。

PG 可以选中任意一个无法从其他箱子所在地到达的箱子所在地作为起点，将任意箱子所在地作为终点，给你堡垒遗迹内所有的箱子所在地及敌对生物信息，请你为 PG 规划一下最优路线，输出 PG 能获得的最大矿藏值。

Input

第一行两个正整数 n, m ($1 \leq n \leq 10^5, 0 \leq m \leq 10^5$)， W ($1 \leq W \leq 100$)，分别代表矿藏数量，有向路径数量和 PG 最大可承受的敌对生物数量。

第二行包含 n 个正整数，第 i 个整数 a_i ($0 \leq a_i \leq 10^9$) 代表第 i 个箱子的下界合金资源数。

接下来 m 行，每行包含三个整数 x, y, w ($1 \leq x, y \leq n$) ($0 \leq w \leq 100$)，代表 x 箱子所在地到 y 箱子所在地有一条有向路径，有 w 单位敌对生物把守。

Output

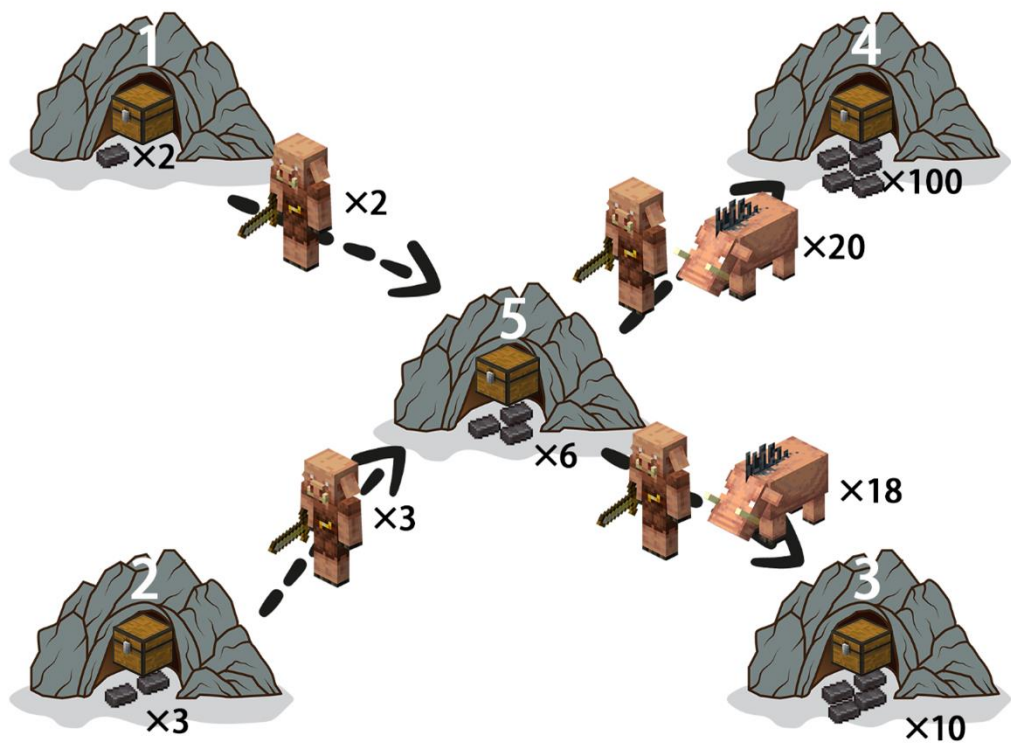
输出 PG 可获得的最大矿藏值。

Examples

standard input	standard output
5 4 21 2 3 10 100 6 1 5 2 2 5 3 5 3 18 5 4 20	19

Explanations

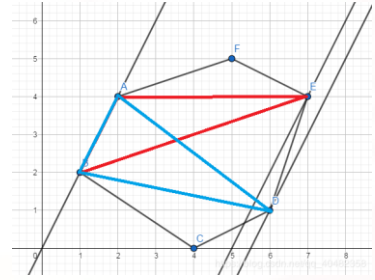
堡垒遗迹地图如下：



显然，PG 选择箱子 2 作为起点，途径 5 到达 3，然后结束挖矿，可获得最大矿藏值 $3 + 6 + 10 = 19$ 。

Problem E. Rotating Calipers

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



Jyo 在上程序设计与 ACM 竞赛公选课的时候睡着了。他梦到了旋转卡壳算法，但是他并没有学过这种算法，只能顾名思义，因此他梦中的旋转卡壳算法可以解决下面的问题：

给你一个多边形，以及多边形内一点作为转轴。将多边形绕转轴旋转，若能被一个该多边形外的固定点“卡住转不动了”（存在某一角度，旋转该角度后，固定点在该多边形的一条边上），则称该多边形在该固定点旋转卡壳。

熟悉计算几何的你当然明白，旋转卡壳算法根本不是用来解决这个问题的，但是你还是需要解决 Jyo 梦到的这个问题。

Input

第一行两个正整数 n, q ($3 \leq n, q \leq 3 \times 10^5$)，分别代表多边形顶点数量以及询问个数。

第二行两个正整数 x, y ($1 \leq x, y \leq 10^9$)，代表转轴坐标 (x, y) 。

接下来 n 行，每行有两个正整数 x, y ($1 \leq x, y \leq 10^9$) 顺时针给出多边形的顶点坐标。

接下来 q 行，每行有两个正整数 x, y ($1 \leq x, y \leq 10^9$) 代表固定点坐标。

Output

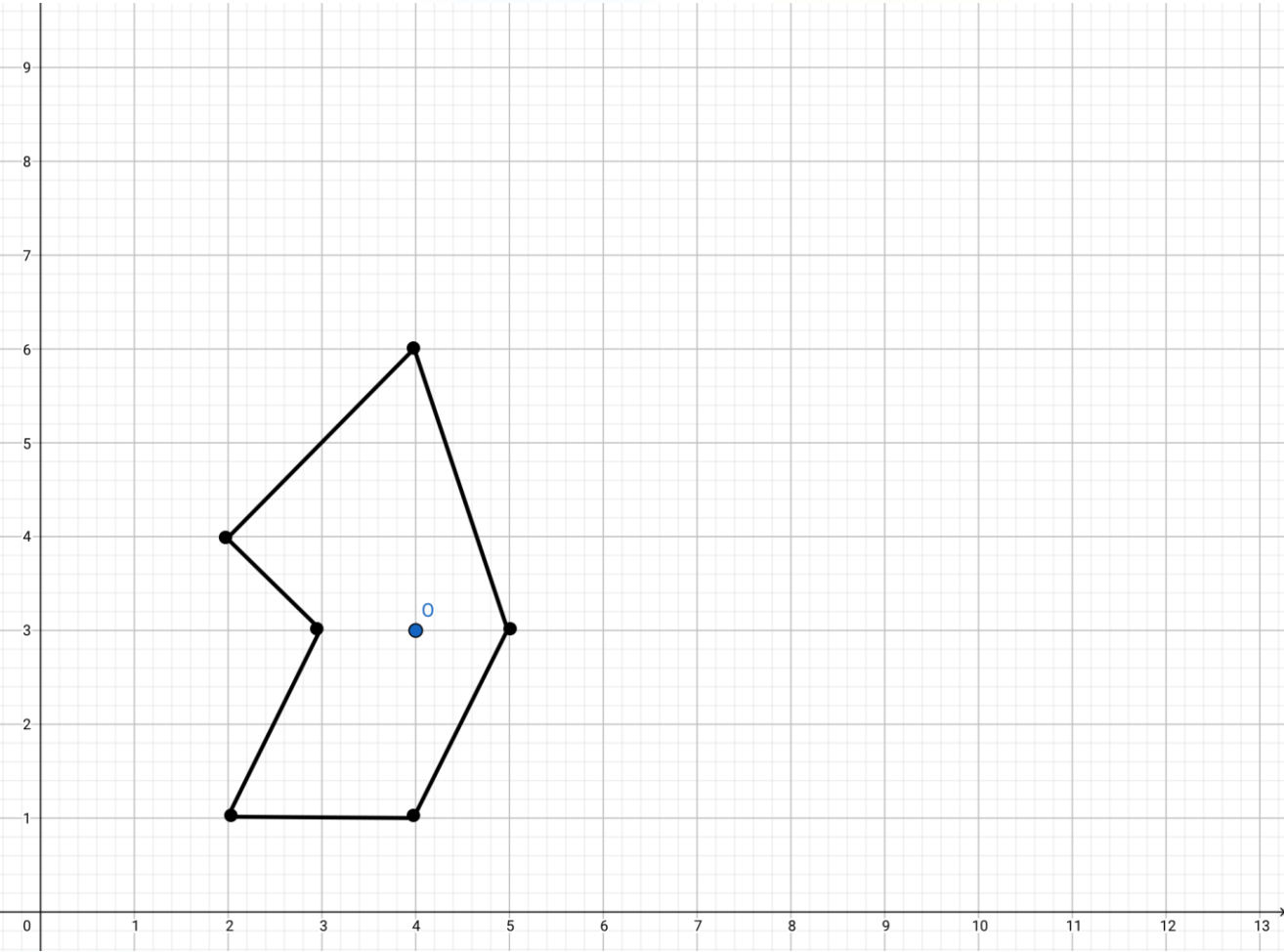
对于每个询问，若多边形在询问给出的固定点旋转卡壳，则输出 YES，否则输出 NO。

Examples

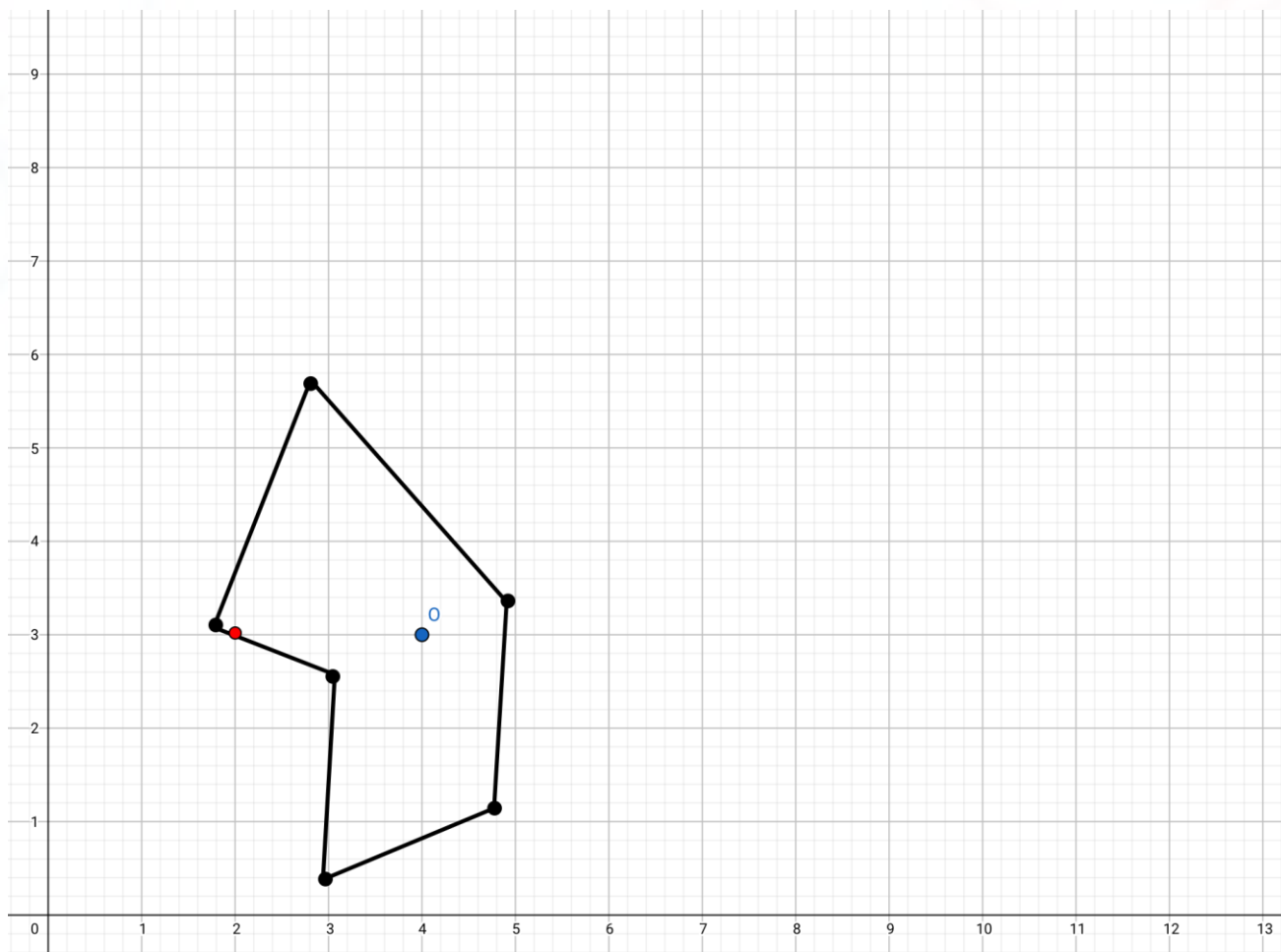
standard input	standard output
6 3	YES
4 3	NO
2 1	YES
3 3	
2 4	
4 6	
5 3	
4 1	
2 3	
5 7	
5 5	

Explanations

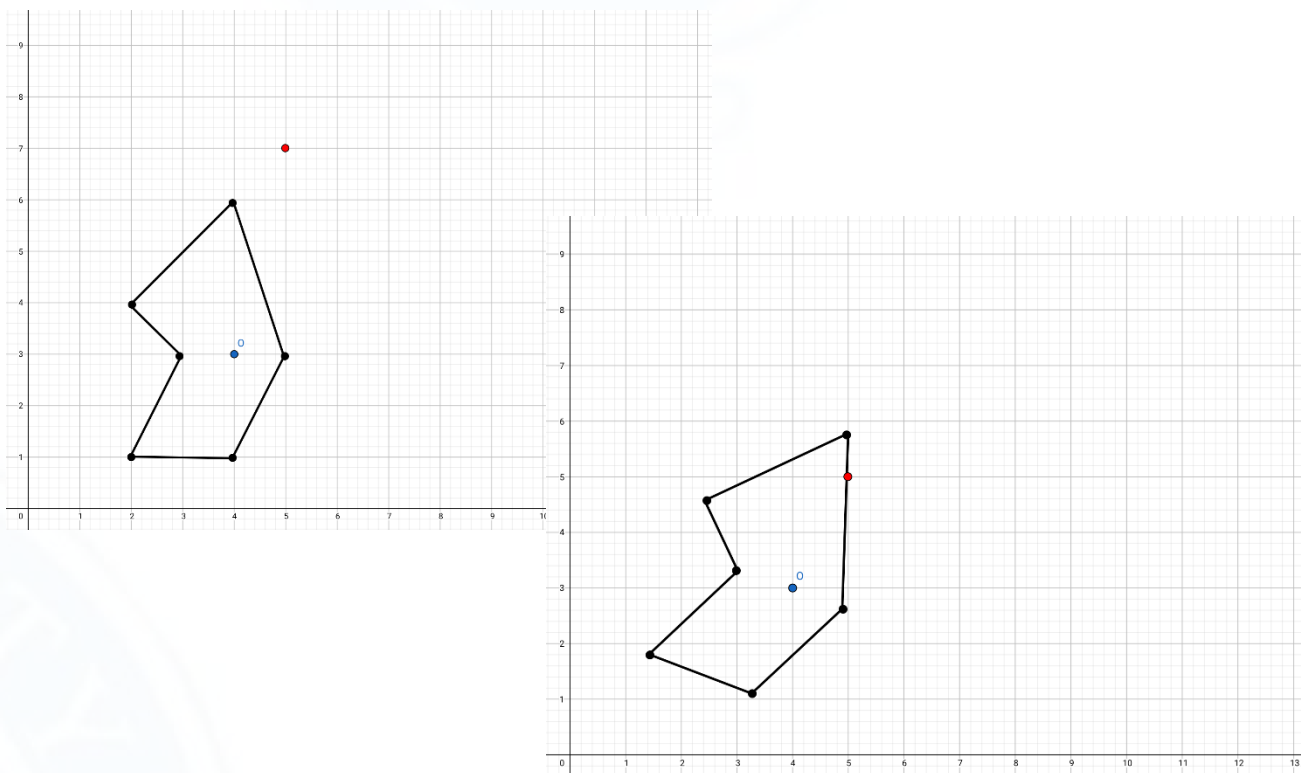
该多边形初始情况如下：



第一个询问：如图所示，多边形在该固定点 (2,3) 旋转卡壳。



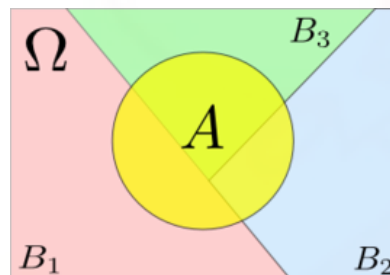
第二、三个询问的情况见下图：



This page is intentionally left blank.
此页有意留为空白。

Problem F. Concise description

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



从 $[1, n]$ 中**等概率**选取两个正整数 a 和 b , 求 $P(\min(a, b) | \max(a, b))$ 。

注明: $|$ 为整除符号, $a | b$ 等价于 $b \% a = 0$ 。

Input

多组测试样例, 输入第一行包含一个数字 $T (1 \leq T \leq 100)$, 表示样例一共有 T 组。

接下来 T 行, 每一行仅包含一个正整数 $n (1 \leq n \leq 10^9)$, 含义见题面。

Output

对于每一个样例, 输出两个正整数中较小数整除较大数的概率, **结果保留八位小数**。

Examples

standard input	standard output
1	0.61111111
6	

Explanations

共有以下 36 种基本事件:

(1,1)	(2,1)	(3,1)	(4,1)	(5,1)	(6,1)
(1,2)	(2,2)	(3,2)	(4,2)	(5,2)	(6,2)
(1,3)	(2,3)	(3,3)	(4,3)	(5,3)	(6,3)
(1,4)	(2,4)	(3,4)	(4,4)	(5,4)	(6,4)
(1,5)	(2,5)	(3,5)	(4,5)	(5,5)	(6,5)
(1,6)	(2,6)	(3,6)	(4,6)	(5,6)	(6,6)

其中标记的是符合条件的基本事件, 共 22 种, 因此概率为 $P(\min(a, b) | \max(a, b)) = \frac{22}{36} \approx 0.61111111$ 。

This page is intentionally left blank.
此页有意留为空白。

Problem G. Diamond Forest

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



Jyo 误入了一片钻石林，这片林子由树木，石头以及钻石组成。

现在 Jyo 拿到了一张地图，地图上有很多连通块。他惊讶的发现：对于一个大小为 n 的连通块，若其中有 $n - 1$ 条边，则该连通块是一棵树；若其不是树，但**任选两个点都有边相连**，则其是一颗钻石；否则该连通块代表一块石头。

请你写个程序，帮助 Jyo 查明这片钻石林里树木，石头以及钻石的数量。

Input

第一行为一个正整数 n, m ($1 \leq n \leq 2 \times 10^5, 1 \leq m \leq \min(2 \times 10^5, \frac{n(n-1)}{2})$), 分别代表地图上的顶点数量和无向边的数量。

接下来 m 行，每行包含两个数字 a, b ($1 \leq a, b \leq n$), 表示顶点 a 和顶点 b 之间有一条无向边。

输入保证不包含重边及自环。

Output

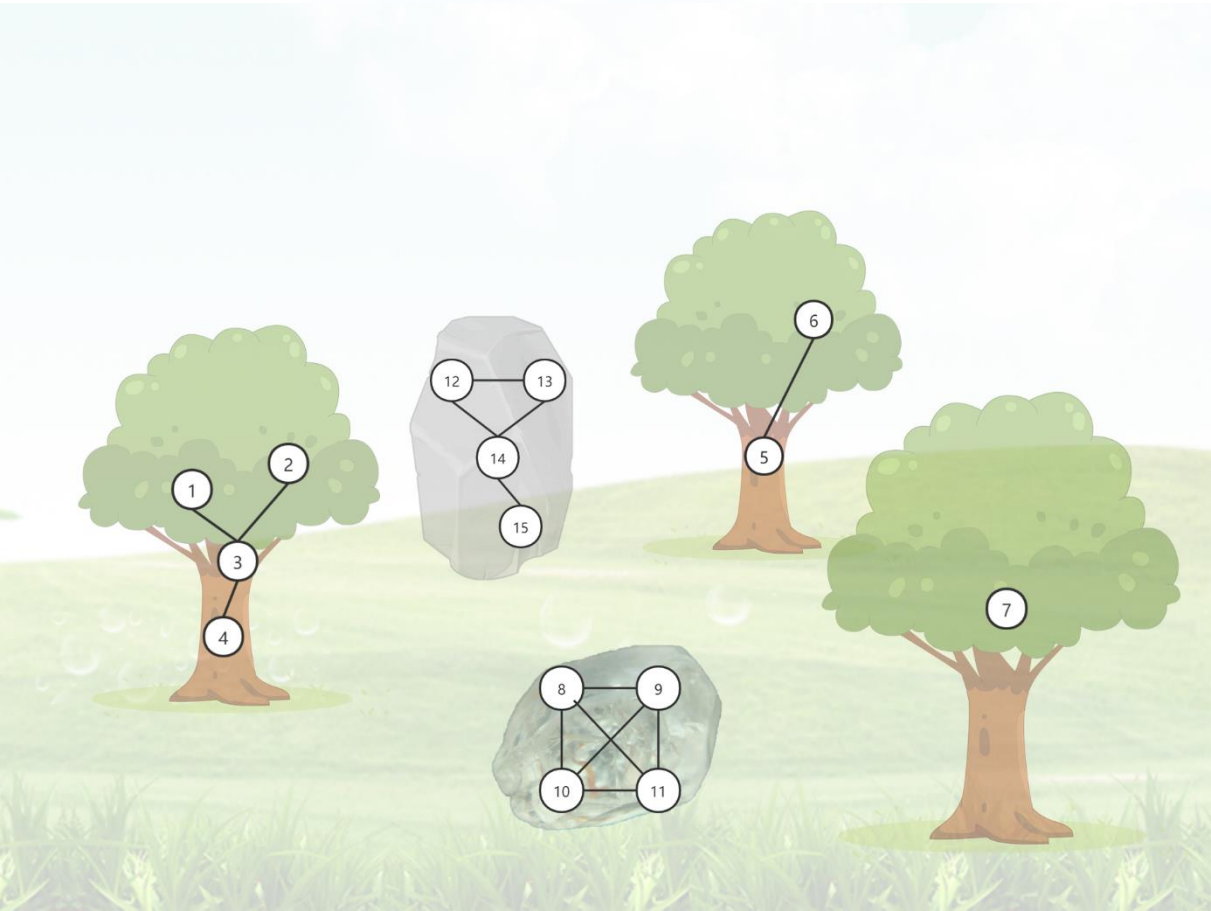
输出三个正整数，分别是树木，石头以及钻石的数量。

Examples

standard input	standard output
15 14 1 3 2 3 3 4 5 6 8 9 10 9 11 10 11 8 8 10 11 9 12 14 13 14 14 15 12 13	3 1 1

Explanations

样例所给情况如下图所示，共有 3 棵树木，1 颗钻石和 1 颗石头。



Problem H. Palindrome Mahjong

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



PG 喜欢打麻将，现在他抽到了一副牌，每张牌可以用一个小写字母表示，因此这副牌可以用小写字符串表示。

现在 PG 可以进行以下两种操作 **各至多一次**：

1. 抽出一张牌，将其放到整副牌的最右边。
2. 选中一张牌，将其变为 PG 想要的任何一张牌。

你需要回答 PG 能否通过操作，将这副牌变为回文。

回文：对于字符串 s ，记该字符串的翻转字符串为 $R(s)$ ，若 $s = R(s)$ ，则称该字符串为回文。如 "abba"、"abccba" 是回文，"abcde"、"ababab" 则不是回文。

Input

多组测试样例，第一行包含一个正整数 T ($1 \leq T \leq 100$)，代表一共有 T 组测试样例。

对于每个测试样例，第一行为一个正整数 n ($1 \leq n \leq 2 \times 10^5$)，代表这副牌的长度。

第二行为一个仅包含小写字母的字符串，代表这副牌。输入保证 $\sum n \leq 2 \times 10^5$ 。

Output

若通过操作，这副牌所代表的字符串可以变为回文字符串，则输出 YES，否则输出 NO。

Examples

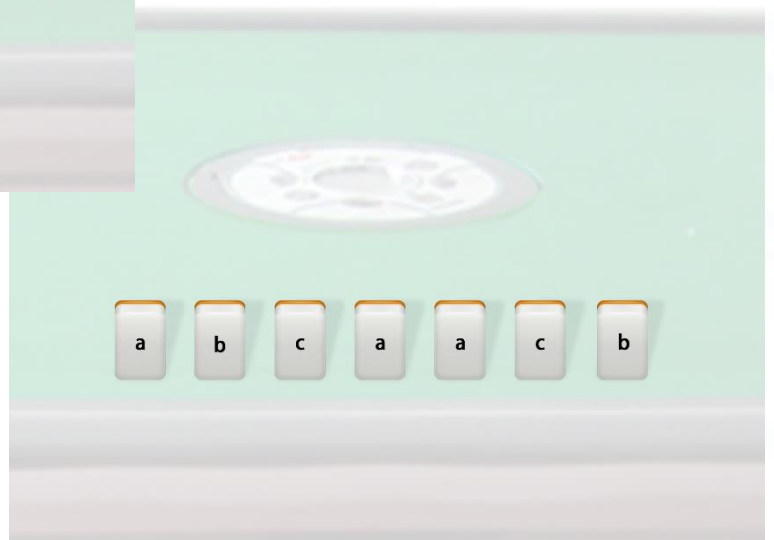
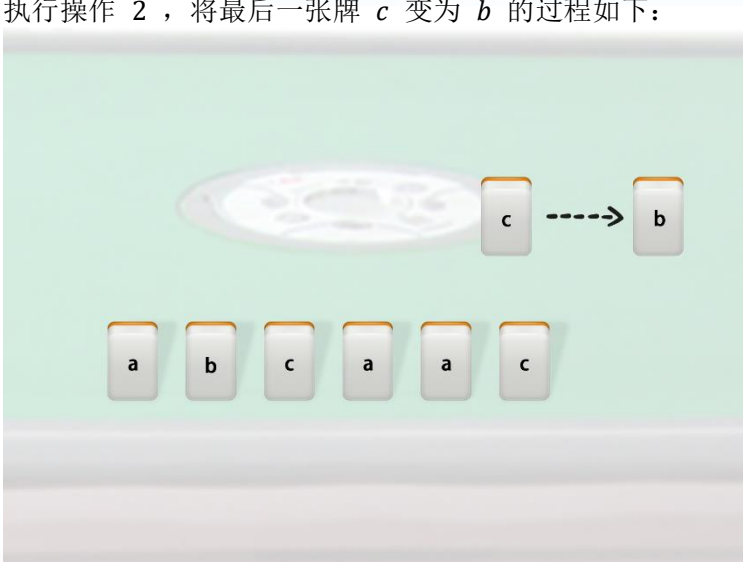
standard input	standard output
1 7 abcaacc	YES

Explanations

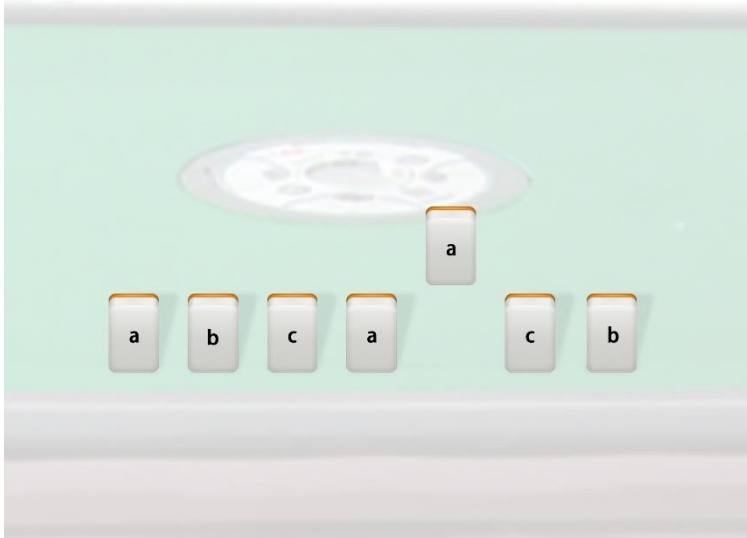
这副牌的初始状态如下图：



执行操作 2，将最后一张牌 c 变为 b 的过程如下：



选中第五张牌，执行操作 1，将其放到整副牌最右边：



最后得到的整副牌为："abcacba"，显然这是回文，因此输出 YES。

This page is intentionally left blank.
此页有意留为空白。

Problem I. The Hand

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes



虹村亿泰的替身是“轰炸空间”，其右手拥有着能削去空间的能力，能够削去一切物质。被削去的东西就像是一开始就不存在一样，也就是削去中间，剩下两端会完美接合。

他走在路上，看到了一串仅由小写字母构成的字符串。此时他想要利用替身能力，删除字符使得字符串的字典序最小，同时为了不破坏市容，他只能发动一次替身能力，即他只能删除一个字符。

但是他并不知道该删哪个字符，请你帮帮他吧。

字典序：原意是表示英文单词在字典中的先后顺序，在计算机领域中扩展成两个任意字符串的大小关系。对于两个字符串，比较两字符串第一个不同的字母，较小的字符串字典序也较小；若较短字符串是较长字符串的前缀，则较短字符串的字典序较小。

Input

多组测试样例，输入第一行包含一个数字 T ($1 \leq T \leq 100$)，表示样例一共有 T 组。

对于每个测试样例，第一行为一个正整数 n ($2 \leq n \leq 2 \times 10^5$)，代表字符串长度。

第二行为一个仅包含小写字母的字符串。输入保证 $\sum n \leq 2 \times 10^5$ 。

Output

对于每一个样例，输出删除一个字符后，能得到的最小字典序的字符串。

Examples

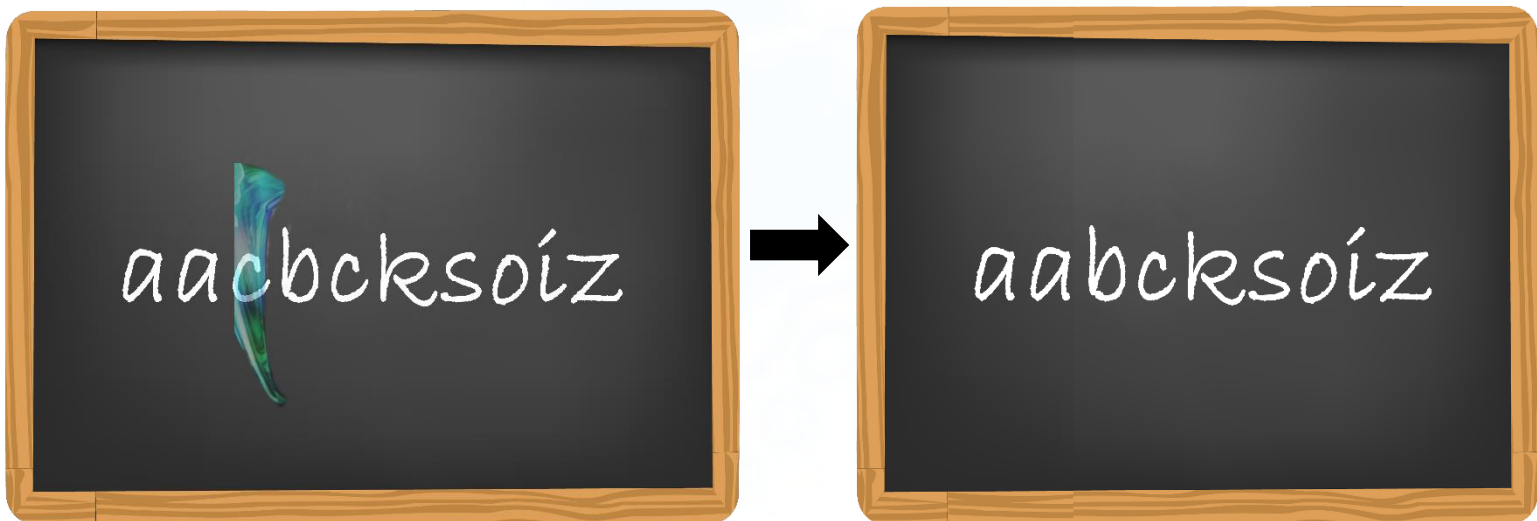
standard input	standard output
1 10 aackbcksoiz	aabcksoiz

Explanations

原始字符串如下图所示：



可以证明，经过删除第三个字母 c，能够获得字符串 "aabcksoiz"，这是经过删除操作后能获得的字典序最小的字符串：



Problem J. Befriend distant states while attacking those nearby

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes



公元前 266 年，秦昭王驱逐了擅权的魏冉，任用范雎为相，并积极推行范雎的“远交近攻”策略。远交近攻先把斗争重点放在近旁的韩赵，因为韩赵两国最近，国力最弱。对较远的齐燕暂置不顾，稳住楚国和魏国两个巨无霸。从而“得寸即王之寸，得尺亦王之尺”。

你穿越成为了两千多年前的范雎，熟读史书的你对此胸有成竹。但是你惊讶地发现，居然有不止七个国家存在于当朝！事实上，一共有 n 个国家，且 n 个国家可以用一维数组 a 表示，其中第 i 个国家的战力值为 a_i 。

所有国家都实行了远交近攻的战略方案：

第 i 个国家会在自己的势力范围 $[\max(1, i - a_i), \min(n, i + a_i)]$ 内，选择一个离自己最远的，且战力值不小于自己战力值的国家建交（若没有符合条件的国家就不建交），因为距离自己较远，因此所有建交都能成功，同时所有国家也会掌握他国的建交关系。

在所有建交完成后，所有国家会在自己的势力范围内选择一个离自己最近的，且战力值小于自己战力值的国家出兵（若没有符合条件的国家就不出兵）。但出兵是有先后顺序的，如果一个国家被一个战力值大于自己的国家出兵，那它就会灭亡；灭亡后的国家自然不能出兵其他国家了。

所有国家遵循以下原则：

1. 如果在相同距离有两所符合条件的国家，无论是建交还是出兵，都优先选择下标最小的国家。
2. 若你本来想出兵的国家建交对象不是你，且战力值不小于你，由于情报通讯不发达，无论该国家建交对象是否存活，你都会放弃出兵该国家。
3. 若你本来想出兵的国家建交对象是你，之后你仍然会出兵该国家。

你现在得到了所有国家的战力值，但是并不知道各个国家的出兵顺序（任何两个国家都不会同时出兵）。对于所有可能的情况，请你计算出每个国家出兵一次（若没有符合条件的国家就不出兵）以后，最小和最大的存活国家数量。

Input

第一行为一个正整数 n ($8 \leq n \leq 2 \times 10^5$), 代表国家数量。

第二行为长度为 n 的数组 a , 其中第 i 个数字 a_i ($1 \leq a_i \leq n$) 代表第 i 个国家的战力值。

Output

请你输出一轮出兵以后, 最小和最大的存活国家数量。

Examples

standard input	standard output
8 1 2 3 4 5 6 7 8	6 6
8 4 3 2 5 7 6 1 3	5 6
8 7 7 7 4 2 6 1 5	7 7

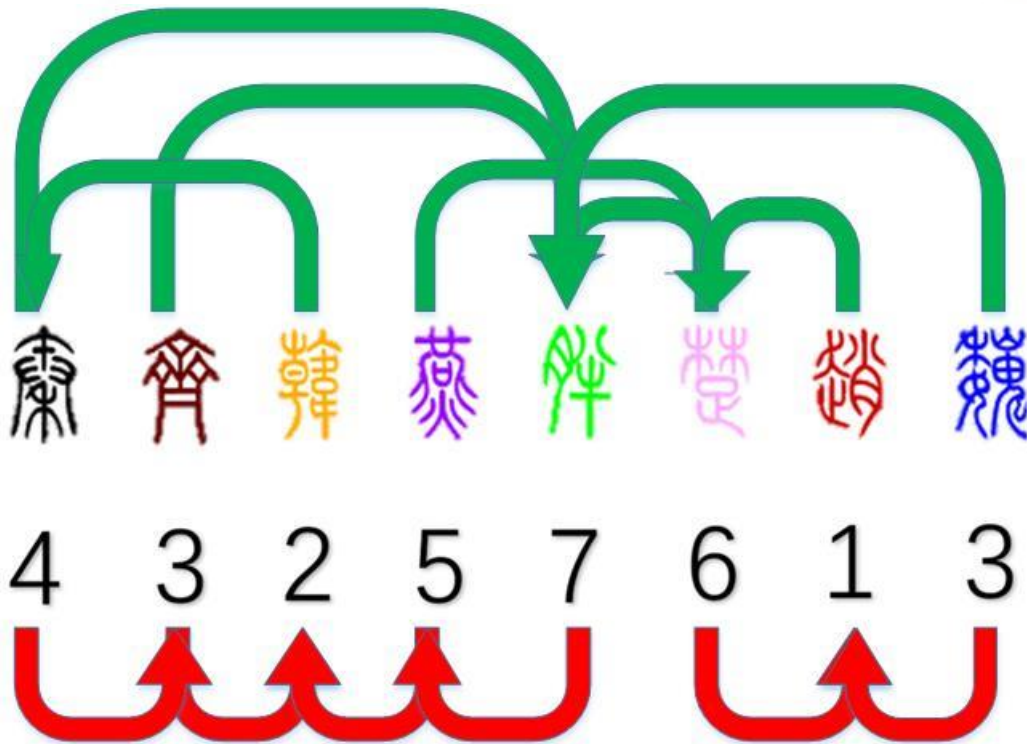
Explanations

样例 2 初始状态如下, 八个国家依次是: 秦, 齐, 韩, 燕, 胖, 楚, 赵, 魏

秦 齐 韩 燕 胖 楚 赵 魏

4 3 2 5 7 6 1 3

根据题意中的远交近攻策略，用绿色箭头代表建交，红色箭头代表出兵，则有：



秦国的范围为 $[1,5]$ ，在势力范围内，最远的不小于自己战力值的国家为胖国，所以其与胖国建交；最近的小于自己战力值的国家为齐国，所以其出兵齐国。

齐国的范围为 $[1,5]$ ，在势力范围内，最远的不小于自己战力值的国家为胖国，所以其与胖国建交；最近的小于自己战力值的国家为韩国，所以其出兵韩国。

韩国的范围为 $[1,5]$ ，在势力范围内，最远的不小于自己战力值的国家为胖国和秦国，所以其与下标更小的秦国建交；势力范围内没有小于自己战力值的国家，因此韩国不出兵。

燕国的范围为 $[1,8]$ ，在势力范围内，最远的不小于自己战力值的国家为楚国，所以其与楚国建交；最近的小于自己战力值的国家为韩国，所以其出兵韩国。

胖国的范围为 $[1,8]$ ，在势力范围内没有不小于自己战力值的国家，因此胖国不建交；最近的小于自己战力值的国家为燕国，所以其出兵燕国。

楚国的范围为 $[1,8]$ ，在势力范围内，最远的不小于自己战力值的国家为胖国，所以其与胖国建交；最近的小于自己战力值的国家为赵国，所以其出兵赵国。

赵国的范围为 [6,8]，在势力范围内，最远的不小于自己战力值的国家为楚国，所以其与楚国建交；势力范围内没有小于自己战力值的国家，因此赵国不出兵。

魏国的范围为 [5,8]，在势力范围内，最远的不小于自己战力值的国家为胖国，所以其与胖国建交；最近的小于自己战力值的国家为赵国，所以其出兵赵国。

若出兵的时间顺序为 韩 - 赵 - 秦 - 齐 - 魏 - 楚 - 燕 - 胖，则：

韩国和赵国因战力值过低不出兵；秦国欲出兵齐国，但因为忌惮胖国，因此选择不出兵；齐国欲出兵韩国，但因为忌惮秦国，因此选择不出兵；魏国欲出兵赵国，但因为忌惮楚国，因此选择不出兵；赵国欲与楚国建交，但楚国选择出兵灭亡赵国，因此赵国灭亡；燕国出兵韩国，因此韩国灭亡；胖国出兵燕国，因此燕国灭亡。最终韩国、燕国及赵国灭亡，因此还有 5 个国家存活，可以证明这是存活国家数量最少的方案之一。

若出兵顺序为 胖 - 魏 - 齐 - 韩 - 赵 - 楚 - 燕 - 秦，则：

胖国出兵燕国，因此燕国灭亡；魏国欲出兵赵国，但因为忌惮楚国，因此选择不出兵；齐国欲出兵韩国，但因为忌惮秦国，因此选择不出兵；韩国和赵国因战力值过低不出兵；赵国欲与楚国建交，但楚国选择出兵灭亡赵国，因此赵国灭亡；燕国已经灭亡，因此无法出兵；秦国欲出兵齐国，但因为忌惮胖国，因此选择不出兵。最终燕国及赵国灭亡，因此还有 6 个国家存活，可以证明这是存活国家数量最多的方案之一。

Problem K. TARDIS

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes



TARDIS 是英国科幻电视剧 Doctor Who 中的时间机器和宇宙飞船，是在宇宙中横冲直撞 (Totally And Radically Driving In Space) 的缩写。

现在 Doctor 被无骨人压入了二维空间，需要从当前位置赶往出口。 Doctor 驾驶着 TARDIS，可以花费一单位阿特隆能量，从上下左右四个方向中选择一个方向移动任意距离。现在给你 Doctor 所在的坐标以及出口坐标，请输出 Doctor 最少需要消耗多少单位阿特隆能量才可以到达出口？

Input

输入仅一行，依次为四个整数 a, b, x, y ($-10^9 \leq a, b, x, y \leq 10^9$)，代表 Doctor 目前的坐标 (a, b) 以及出口坐标 (x, y) 。

Output

输出一个整数，代表 Doctor 到达出口最少需花费的阿特隆能量单位。

Examples

standard input	standard output
2 3 4 5	2
1 4 1 109	1

Explanations

对于样例 1，Doctor 目前在坐标 $(2,3)$ 处，他可以先花费一单位能量移动到坐标 $(2,5)$ 处，然后再花费一单位能量移动到出口 $(4,5)$ ，可以证明这样移动所耗费的能量最少。

对于样例 2，Doctor 只需要花费一单位能量就可以直接到达出口，因为所在位置与出口在同一直线上。

This page is intentionally left blank.
此页有意留为空白。

Problem L. Nim Type Zero

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



露娜要和你玩两人式的尼姆零式扑克牌游戏，规则是这样的：用「0」，「1」，「2」，「3」这四种牌各十张，共 40 张，每个人每次发四张作为手牌，两个人按顺序出牌，场上的数字超过 9 的话，就是出牌的那个人输了。

每局输了的人可以获得 1 分，赢了的人可以获得 2 分。露娜赢少输多，于是私自篡改了很多得分数据。你一时无从分辨哪些得分是可能出现的情况，因此你决定写一个程序帮助你识别哪些得分是可能出现的。

Input

多组测试样例，输入第一行包含一个数字 T ($1 \leq T \leq 5 \times 10^5$)，表示样例一共有 T 组。

接下来 T 行，每一行包含两个整数 a, b ($0 \leq a, b \leq 10^{16}$)，分别代表你的得分以及露娜的得分。

Output

对于每一个样例，若得分是可能出现的，则输出 YES，否则输出 NO。

Examples

standard input	standard output
4	YES
0 0	NO
4 4	YES
7 5	YES
1000 2000	

Explanations

对于第一个样例 4 4，可以证明无论进行多少局游戏，胜负关系如何，得分 4 4 都不会出现。

对于第二个样例 7 5，若共进行 4 局游戏，你赢 3 局，露娜赢 1 局，则你共获得 $3 \times 2 + 1 \times 1 = 7$ 分，而露娜共获得 $1 \times 2 + 3 \times 1 = 5$ 分，所以得分 7 5 是可能出现的。

对于第三个样例 1000 2000，若共进行 1000 局游戏，你未尝一胜，则你共获得 $0 \times 2 + 1000 \times 1 = 1000$ 分，而露娜共获得 $1000 \times 2 + 0 \times 1 = 2000$ 分，所以得分 1000 2000 是可能出现的。