

Problem A. 只有一位数字

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

给你一个整数 x 。您需要找到最小的非负整数 y ，使得数字 x 和 y 至少共享一个共同数字。换句话说，必须存在一个十进制数字 d 同时出现在数字 x 和数字 y 的表示中。

Input

第一行包含一个整数 t ，表示测试用例的数量。

每个测试用例的第一行包含一个整数 x 。

Output

对于每个测试用例，输出一个整数 y —— 满足条件的最小非负整数。

Example

standard input	standard output
5	6
6	6
96	7
78	1
122	6
696	

Note

$1 \leq t \leq 1000, 1 \leq x \leq 1000$

Problem B. gmk与质数

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

gmk 对质数情有独钟。最近，他发现了一种有趣的数字生成算法：

给定一个整数 x （无前导零）和一个重复次数 k 。将 x 的十进制表示重复 k 次，得到一个新的整数 y 。

例如，当 $x = 52, k = 3$ 时， $y = 525252$ 。当 $x = 6, k = 7$ 时， $y = 6666666$ 。

gmk 非常希望这样生成的数 y 是一个质数。请你编写一个程序帮助 gmk 判断 y 是否为质数。

* 质数是指在大于 1 的自然数中，除了 1 和它本身以外不再有其他因数的自然数。注意，1 不是质数。

Input

第一行包含一个整数 t ，表示测试用例的数量。

接下来 t 行，每行包含两个整数 x 和 k 。

Output

对于每个测试用例，如果生成的数 y 是质数，则输出 ‘YES’，否则输出 ‘NO’。

Example

standard input	standard output
4	NO
52 3	NO
6 7	YES
7 1	NO
1 7	

Note

$1 \leq t \leq 100, 1 \leq x \leq 10^9, 1 \leq k \leq 7$

Problem C. 平均水平

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

您将获得一个长度为 n 的数组 a 。

您的任务是找到数组 a 的任何子数组的最大可能平均值。

形式上, 对于任何满足 $1 \leq l \leq r \leq n$ 的索引 l, r , 将子数组 a_l, a_{l+1}, \dots, a_r 的平均值定义为元素之和除以元素数量, 或者:

$$\text{avg}(l, r) = \frac{1}{r - l + 1} \sum_{i=l}^r a_i$$

输出 l, r 所有选项中 $\text{avg}(l, r)$ 的最大值。

* 如果可以通过删除开头的几个 (可能是零个或全部) 元素以及从末尾删除几个 (可能是零个或全部) 元素来从 a 获得 b , 则数组 b 是数组 a 的子数组。特别是, 数组是其自身的子数组。

Input

第一行包含一个整数 t , 表示测试用例的数量。

每个测试用例的第一行包含一个整数 n , 表示数组 a 的长度。

每个测试用例的第二行包含 n 个整数 a_1, a_2, \dots, a_n , 表示数组的元素。

Output

对于每个测试用例, 输出一个整数, 表示给定数组的任何子数组的最大平均值。

* 可以证明答案总是一个整数。

Example

standard input	standard output
3	3
4	9
3 3 3 3	4
5	
7 1 6 9 9	
5	
3 4 4 4 3	

Note

$1 \leq t \leq 10^4, 1 \leq n \leq 10, 1 \leq a_i \leq 10$ 。

Problem D. 数字游戏 I

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

你拿到一大堆数字卡片。规则是：这里面只有两张卡片，我们叫它们 a 和 b ，它们的数量是奇数张（比如 1 张、3 张、5 张...）。除了它俩，其他所有种类的卡片都保证是偶数张（比如 2 张、4 张、6 张...）。你的任务就是找出 a 和 b 是谁。

形式化地：在多重集 S 中，存在且仅存在两个不同的元素 a 和 b ，它们各自的出现次数为奇数。所有其他元素的出现次数均为偶数。

Input

一个整数 T ，表示测试用例的数量。

对于每个测试用例：

一个整数 n ，表示多重集（数组）的大小。

一个包含 n 个整数的多重集 $S = \{s_1, s_2, \dots, s_n\}$ 。

Output

对于每个测试用例，输出一行，包含两个整数 a 和 b ，按升序排列，并用单个空格分隔。

Example

standard input	standard output
3	-5 -2
12	6 10
-6 -5 -5 -2 -5 -5 -6 -5 -5 -2 -5 -2	-10 3
14	
7 6 6 10 6 10 9 9 10 10 10 7 10	
12	
9 3 10 8 5 -10 3 3 10 8 5 9	

Note

$1 \leq T \leq 10$, $3 \leq n \leq 2 \times 10^5$, $-10^5 \leq s_i \leq 10^5$ 。

Problem E. 国王逃脱

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

在一个 $n \times n$ 的棋盘上, 爱丽丝的皇后位于 (a_x, a_y) , 鲍勃的国王位于 (b_x, b_y) 。爱丽丝不再移动棋子, 只有鲍勃可以移动他的国王。

国王每次可以移动到周围八个相邻的格子中的任意一个。如果国王所在的格子与皇后在同一行、同一列或同一对角线, 则国王被将军。

鲍勃的目标是将他的国王从起点 (b_x, b_y) 移动到终点 (c_x, c_y) , 且移动路径中的每一步都不能被将军。请判断鲍勃能否成功。

Input

第一行一个整数 n , 表示棋盘大小。

第二行两个整数 a_x, a_y ($1 \leq a_x, a_y \leq n$), 表示皇后的坐标。

第三行两个整数 b_x, b_y ($1 \leq b_x, b_y \leq n$), 表示国王的起始坐标。

第四行两个整数 c_x, c_y ($1 \leq c_x, c_y \leq n$), 表示国王的目标坐标。

* 数据保证: 国王的起点和终点均未被将军, 且后、王、目标点三点互不重合。

Output

如果鲍勃可以成功, 输出 ‘YES’。否则, 输出 ‘NO’。

Examples

standard input	standard output
8 4 4 1 3 3 1	YES
8 4 4 2 3 1 6	NO
8 3 5 1 2 6 1	NO

Note

$3 \leq n \leq 1000$, $1 \leq a_x, a_y, b_x, b_y, c_x, c_y \leq n$ 。

Problem F. 数字游戏 II

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

桌面上摆放着 n 张数字卡片, n 是一个偶数。Alice 和 Bob 将进行 $n/2$ 轮游戏来取走这些卡片。

每一轮的行动顺序如下:

1. Alice 从桌面上任选一张卡片, 并将其取走。
2. Bob 从桌面上剩余的卡片中任选一张, 并将其取走。

计分规则

在每一轮中, 如果 Alice 取走的卡片数字 a 与 Bob 取走的卡片数字 b 之和能被 9 整除 (即 $(a + b) \pmod{9} = 0$), 那么 Bob 获得 1 分, 否则不得分。

玩家目标

Bob 的目标是最大化自己的总得分。Alice 的目标是最小化 Bob 的总得分。

假设两名玩家都采取最优策略。请计算游戏结束时, Bob 的最终得分。

Input

第一行是一个整数 t , 表示测试用例的数量。

每个测试用例包含两行:

第一行是一个偶数 n 。

第二行是 n 个正整数 a_1, a_2, \dots, a_n , 代表卡片上的数字。

Output

对于每个测试用例, 输出一个整数, 表示 Bob 的最终得分。

Example

standard input	standard output
3	2
4	4
46 29 16 35	0
8	
18 33 7 24 18 39 38 48	
12	
19 28 12 39 12 37 46 20 10 29 20 20	

Note

$1 \leq t \leq 1000, 2 \leq n \leq 1000, 1 \leq a_i \leq 10^5$

Problem G. gmk和121

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

gmk 有一个由 n 个整数组成的数组 a , 元素编号从 1 到 n 。
在一项操作中, 他可以选择索引 i ($2 \leq i \leq n - 1$) 并分配:

- $a_{i-1} = a_{i-1} - 1$
- $a_i = a_i - 2$
- $a_{i+1} = a_{i+1} - 1$

gmk 可以多次应用此操作。任何索引 i 都可以使用零次或多次。
他能用这个操作使数组的所有元素都等于 0 吗?

Input

输入的第一行包含一个整数 t , 表示测试中的测试用例数。
每种情况的第一行包含一个整数 n , 表示数组中的元素数量。
每种情况的第二行包含 n 个整数 a_1, a_2, \dots, a_n , 表示数组的元素。

Output

对于每个测试用例, 如果可以使用所描述的操作使数组的所有元素为零, 则输出 ‘YES’。否则, 输出 ‘NO’。

Example

standard input	standard output
7	YES
5	NO
1 3 5 5 2	YES
5	NO
2 4 4 5 1	NO
5	NO
0 1 3 3 1	NO
6	
5 6 0 2 3 0	
4	
1 2 7 2	
3	
7 1 0	
4	
1 1 1 1	

Note

$1 \leq t \leq 10^4$, $3 \leq n \leq 2 \cdot 10^5$, $0 \leq a_j \leq 10^9$ 。
保证所有测试用例的 n 值之和不超过 $2 \cdot 10^5$ 。

Problem H. 开宝箱咯

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

想象一下，你有一排 n 个宝箱 a_1, a_2, \dots, a_n 。你要按某种顺序把它们全部打开，每次打开一个。当你决定打开第 k 个宝箱（它最初在 k 号位置）时，会发生两件事：

1. 你获得一个数字。这个数字等于宝箱里的财富 a_k 加上它当前的位置 k 。
2. 这个宝箱从队伍里消失，它后面的宝箱都向前挪一步。

你总共会获得 n 个数字，但这些数字需要放进一个集合 S 里。集合的特点是 不允许重复，放进去一个 100，再放一个 100，集合里还是只有一个 100。最后，把集合 S 里的所有数字从大到小排成一队，得到数组 b 。你的目标是，通过设计打开宝箱的顺序，让这个最终的数组 b 的字典序最大。"字典序最大"，通俗讲，就是让 b 的第一个数越大越好；在第一个数最大的前提下，让第二个数越大越好；以此类推。

Input

每个测试包含多个测试用例。第一行包含一个整数 t ，表示测试用例的数量。测试用例的描述如下。

每个测试用例的第一行包含一个整数 n ，表示数组 a 的长度。

每个测试用例的第二行包含 n 个整数 a_1, a_2, \dots, a_n ，表示数组 a 的元素。

Output

对于每个测试用例，输出字典顺序最大的 b 。

Example

standard input	standard output
5	3 2
2	1000000005 1000004 1003 102 2
2 1	11 7 6
5	48 36 31 20 18 17 13 11
1 100 1000 1000000 1000000000	36 30 22 19 16 14 11 10 2
3	
6 4 8	
8	
17 16 28 32 43 14 6 3	
9	
1 8 8 10 11 13 15 22 27	

Note

$$1 \leq t \leq 10^4$$

$$1 \leq n \leq 3 \times 10^5$$

$$1 \leq a_i \leq 10^9$$

所有测试用例的 n 总和不超过 3×10^5 。

Problem I. 相同的出现次数

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

当且仅当任何元素出现的次数相同时，我们称数组平衡。例如， $[1, 1, 3, 3, 6, 6]$ 和 $[2, 2, 2, 2]$ 是平衡的，但 $[1, 2, 3, 3]$ 是不平衡的（元素 1 和 3 出现的次数不同）。请注意，空数组始终是平衡的。

给您一个由 n 整数组成的非递减数组 a 。求其最长平衡子序列的长度。

如果可以通过从任意位置删除多个（可能是零个或全部）元素从 a 获得 b ，则序列 b 是序列 a 的子序列。

Input

每个测试包包含多个测试用例。第一行包含测试用例的数量 t 。测试用例的描述如下。

每个测试用例的第一行包含一个整数 n ，表示 a 的长度。

第二行包含 n 整数 a_1, a_2, \dots, a_n ，表示 a 的元素。

Output

对于每个测试用例，输出一个整数 a 的最长平衡子序列的长度。

Example

standard input	standard output
4	4
5	2
1 1 4 4 4	9
2	5
1 2	
15	
1 1 1 1 1 2 2 2 2 3 3 3 4 4 5	
5	
3 3 3 3 3	

Note

$1 \leq t \leq 500, 1 \leq n \leq 100, 1 \leq a_1 \leq a_2 \leq \dots \leq a_n \leq n$ 。

Problem J. 泉信也玩 ACM

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

泉州信息工程学院（Quanzhou University of Information Engineering）学校坐落于福建省泉州市，受福建省教育厅管辖，是经中华人民共和国教育部批准成立的民办全日制应用型本科高校，是全国第一批教育信息化试点优秀单位、第六届全国黄炎培职业教育优秀学校、福建省高校“三全育人”综合改革试点建设单位，为福建省硕士学位授予培育单位、福建省一流应用型本科高校培育单位。

学校创办于 2002 年，前身为泉州信息职业技术学院；2004 年 4 月，经福建省人民政府批准创办、具有独立颁发国家承认学历的一所普通高等院校。2009 年 5 月，被确定为首批实施福建省“闽台高职院校—校—企联合培养人才”项目试点校；12 月，通过教育部人才培养工作评估。2014 年教育部批准同意在泉州信息职业技术学院的基础上建立泉州信息工程学院；2018 年成为学士学位授予权单位。2021 年顺利通过教育部本科教学工作合格评估，被列为硕士学位授予培育单位；2022 年入选福建省一流应用型本科高校培育单位。

截至 2023-2024 学年，学校总占地面积 100.67 万平方米，总建筑面积为 43.79 万平方米。有教学、科研仪器设备资产总值 1.61 亿元。截至 2024 年 7 月，学校设有 10 个二级教学单位，开设 33 个本科招生专业。有专任教师 900 余人，在校生近 16000 人。截至 2024 年 8 月，馆藏图书 133.25 万册、中外文电子期刊 13.45 万册，电子图书 70.3 万册。截至 2024 年 7 月，学校有国家级一流专业建设点 3 个、省级一流专业建设点 8 个，国家一流课程 1 门。

Input

本题没有输入。

Output

输出 ‘QUIE NO1’。

Problem K. 能量水晶

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

现有三个能量水晶，其编号分别为 $1, 2, 3$ 。我们记第 i 个水晶的能量等级为 a_i 。初始时，所有水晶均未充电，即其能量等级均为 0 。每个水晶都需要被充电至能量等级恰好为 x （不能超过）。

在一次操作中，可以选择任意一个水晶，并将其能量等级增加任意正值。然而，能量水晶之间是相互同步的，一次操作仅当其完成后满足以下条件时才能被执行：

- 对于任意一对水晶 i, j ，必须满足 $a_i \geq \lfloor \frac{a_j}{2} \rfloor$ 。

请问，将所有水晶充电至目标状态所需的最小操作次数是多少？

Input

本题包含多组测试数据。输入的第一行包含一个整数 t ，表示测试数据的组数。接下来是各组测试数据的描述。

每组测试数据仅占一行，包含一个整数 x 。

Output

对于每组测试数据，输出一个整数 —— 将所有能量水晶充电至等级 x 所需的最小操作次数。

Example

standard input	standard output
7	3
1	7
5	9
14	23
2025	31
31415	59
536870910	61
1000000000	

Note

$1 \leq t \leq 10^4, 1 \leq x \leq 10^9$

Problem L. gmk的二进制

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

给定一个长度为 n 的二进制字符串 s 和一个正整数 k 。你可以选择 s 中的一些位置进行"保护"。然后，游戏按顺序从 $i = 1$ 到 n 进行如下操作：

如果 s_i 满足以下所有条件， gmk 可以将 s_i 变为 '0'：

1. $s_i = 1$
2. s_i 未被保护
3. 在 s 的子串 $s_{\max(1, i-k+1)}, \dots, s_{i-1}$ 中没有 '1'（即 s_i 前面的 $k-1$ 个字符都是 '0'）

你的目标是，用最少的保护次数，使得 gmk 无法对字符串 s 作出任何改变。请求出这个最小的保护次数。

Input

每个测试包含多个测试用例。第一行包含测试用例的数量 t 。

每个测试用例的第一行包含整数 n 和 k 。

每个测试用例的第二行包含一个长度为 n 的二进制字符串 s ，由字符 '0' 和 '1' 组成。

Output

对于每个测试用例，输出需要保护的最小位置数，以强制 gmk 保持字符串不变。

Example

standard input	standard output
9	1
2 2	1
11	1
6 6	4
100001	1
5 3	1
10000	1
7 2	2
1010101	0
7 4	
0000001	
3 3	
010	
3 2	
011	
7 4	
1001001	
8 3	
00000000	

Note

$1 \leq t \leq 100$, $2 \leq n \leq 1000$, $2 \leq k \leq n$, 所有测试用例的 n 总和不超过 1000。

Problem M. gmk的序列

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

gmk 有一个整数 x 。他通过交替整数 x 和 $-x$ 创建了一个长度为 n 的序列，从 x 开始。
例如，如果是 $n = 5$ ，则序列如下所示: $x, -x, x, -x, x$ 。
他要求你找出序列中所有整数的总和。

Input

第一行包含一个整数 t ，表示测试用例的数量。
每个测试用例包含两个整数 x 和 n 。

Output

对于每个测试用例，输出序列中所有整数的总和。

Example

standard input	standard output
4	0
1 4	2
2 5	0
3 6	4
4 7	

Note

$1 \leq t \leq 100, 1 \leq x, n \leq 10$ 。