

Problem A. All Kinds of Dice

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

将"骰子"定义为至少有一个面且每个面显示正整数的任何形状。当骰子被掷出时，随机选择其中一个面。当两个骰子相互掷出时，显示更高数字的骰子获得 1 分；如果两个数字相等，则每个骰子获得 1/2 分。对于骰子 D 和 D_0 ，定义 $\text{score}(D, D_0)$ 为 D 在与 D_0 的一次掷骰中获得的期望分数。

给定一个包含 N 个顶点的竞赛图，请构造 N 个骰子 D_1, D_2, \dots, D_N ，使得

- 对于每条有向边 $i \rightarrow j$ ，有 $\text{score}(D_i, D_j) > 1/2$ 。

报告是否可以做到这一点。

Input

输入的第一行包含一个整数 N ($2 \leq N \leq 100$)，表示竞赛图中的顶点数量。

接下来， N 行描述比赛。第 i 行包含一个二进制字符串 s_i 。当且仅当 $s_{i,j}$ 等于 '1' 时，竞赛图中存在有向边 $i \rightarrow j$ 。

保证对于任何 $1 \leq i < j \leq N$ ， $s_{i,j}$ 和 $s_{j,i}$ 中恰好有一个等于 '1'，并且对于所有 $1 \leq i \leq N$ ， $s_{i,i}$ 等于 '0'。

Output

如果不存在解决方案，则在一行中输出 -1。

否则，输出 N 行。第 i 行描述第 i 个骰子：

- 第一个整数 k 表示第 i 个"骰子"的面数；
- 然后跟随 k 个整数 v_1, v_2, \dots, v_k ($1 \leq v_i \leq 10^9$)，表示每个面所写的整数。

所有骰子的面总数应限制在 10^5 以内。

Example

standard input	standard output
4	3 1 5 9
0100	3 3 4 8
0010	3 2 6 7
1001	3 5 5 6
1100	

Problem B. Blind Alley

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

《超级马里奥制造》是为了纪念《超级马里奥》系列30周年而制作的作品。《超级马里奥制造》的最大特点是整个游戏包含一个完整的关卡编辑系统，允许玩家使用Wii U GamePad设计和创建马里奥关卡；玩家还可以将自己设计的关卡上传到任天堂的服务器，与全球玩家分享。

作为《超级马里奥》的忠实玩家，你创建了一个气球关卡！这个关卡可以视为一个有限大小的网格，其中一些单元格包含无法通行的障碍物，例如尖刺。玩家需要控制"气球马里奥"从网格的第一列移动到最后一列，通过相邻的单元格并避免尖刺等障碍物。作为一款横向卷轴游戏，其独特之处在于有限的视野，超出可见范围的区域是无法到达的。这一设定对合理的关卡设计提出了重大挑战：即使对于非常聪明的玩家，视野的限制也可能导致他们进入"死胡同"，无法完成关卡。在这个问题中，你需要确定你设计的关卡中是否存在这种现象。

在这个问题中，我们将考虑以下简化模型。一个关卡将被视为一个大小为 $N \times M$ 的网格，"气球马里奥"角色从 $(1, 1)$ 开始，玩家的目标是将"气球马里奥"操控到 $(1, M)$ 。一些单元格包含障碍物，而"气球马里奥"角色无法通过这些单元格。第一列和最后一列没有放置障碍物。当"气球马里奥"角色位于 (X, Y) 时，玩家的视野为 $\{(U, V) \mid 1 \leq U \leq N, Y \leq V \leq \min(Y + K, M)\}$ ，这意味着玩家可以看到这些单元格中是否有障碍物。当"气球马里奥"角色位于 (X, Y) 时，角色可以向 $(X + 1, Y)$ 上移，向 $(X - 1, Y)$ 下移，或向 $(X, Y + 1)$ 右移，只要目标单元格在地图边界内且不包含障碍物。

你需要确定是否存在一条从起点 $s_0 = (1, 1)$ 到某个点 $s_\ell = (X, Y)$ 的路径，使得以下条件同时满足：

- "气球马里奥"可以沿着路径 $P = s_0 s_1 \dots s_\ell$ 从 s_0 移动到 s_ℓ 。
- 对于任何 $1 \leq i \leq \ell$ ，玩家无法根据在 s_{i-1} 的可见信息排除从 s_i 到达 $(1, M)$ 的可能性。
- 然而，玩家无法从 s_ℓ 将"气球马里奥"移动到 $(1, M)$ 。

Input

输入的第一行包含一个正整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。每个测试用例描述如下。

每个测试用例的第一行包含三个正整数 N, M, K ($2 \leq N, M, N \cdot M \leq 10^6, 1 \leq K \leq M - 1$)，表示地图的大小参数和可见范围参数。接下来的 N 行每行包含一个长度为 M 的二进制字符串 s_i ，表示地图第 i 行的障碍物信息。这里， $s_{i,j} = 1$ 当且仅当在位置 (i, j) 有障碍物。输入保证第一列和最后一列没有障碍物，并且从 $(1, 1)$ 到 $(1, M)$ 存在一条路径。

保证所有测试用例中 $N \cdot M$ 的总和不超过 10^6 。

Output

对于每个测试用例，如果存在满足问题中所述条件的路径，则在一行中输出"是"（不带引号）；否则，输出"否"（不带引号）。评估不区分大小写，即"YES"、"yes"、"YeS"等都被接受。

Example

standard input	standard output
6	Yes
3 5 2	No
00010	No
01110	No
00000	No
3 5 3	Yes
00010	
01110	
00000	
3 5 2	
01000	
01110	
00000	
3 5 2	
00000	
01110	
00000	
3 5 2	
01010	
01110	
00000	
5 5 2	
00000	
00110	
00010	
01010	
00010	

Note

对于第一个测试样例，玩家在 (1,1) 的可见信息可以表示为以下矩阵:

000??

011??

000??

根据这些信息，玩家无法排除从 (1,2) 到达 (1,5) 的可能性。然而，"气球马里奥"无法从 (1,2) 移动到 (1,5)。

对于其他测试样例（最后一个除外），玩家不会因为视野限制而遇到无法到达终点的死胡同。

Problem C. Computational Geometry Problem

Input file: `standard input`
Output file: `standard output`
Time limit: 5 seconds
Memory limit: 1024 megabytes

给定整数 N 、 M 和 K ，计算满足以下约束条件的整数序列 $A = [A_1, A_2, \dots, A_{NM}]$ 的数量：

1. $1 \leq A_i \leq M$ ，且从 1 到 M 的每个数字在 A 中恰好出现 N 次；
2. 不存在整数 $1 \leq i < j < k < \ell \leq NM$ 满足 $A_i = A_k$ ， $A_j = A_\ell$ ，且 $A_i \neq A_j$ ；
3. 满足 $1 \leq i < NM$ 且 $A_i = A_{i+1}$ 的数量为 K 。

答案可能非常庞大，您应该输出答案对 998 244 353 取模的结果。

Input

输入的唯一一行包含三个整数 N 、 M 和 K （ $1 \leq N, M \leq 10^7$ ， $0 \leq K \leq NM - 1$ ）。

Output

输出一个整数，表示答案对 998 244 353 取模的结果。

Examples

standard input	standard output
3 3 6	6
3 3 5	36
5 3 3	0
35 13 433	131803241

Problem D. Determinant of 01-Matrix

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

给定一个正整数 $N \leq 10^9$ ，请构造一个仅包含 0s 和 1s 的矩阵 $A = (a_{i,j})_{1 \leq i,j \leq n}$ ，满足

- $1 \leq n \leq 200$;
- $\det(A) = N$ 。

保证在问题约束下存在有效的解决方案。

Input

输入仅包含一个正整数 N ($1 \leq N \leq 10^9$)。

Output

输出应包含第一行一个正整数 n ($1 \leq n \leq 200$)，表示构造的矩阵的大小。

接下来，输出 n 行，其中第 i 行包含 n 个整数 $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ ($a_{i,j} \in \{0, 1\}$)，表示矩阵第 i 行的元素。如果有多个解决方案，可以输出其中任何一个。

Examples

standard input	standard output
1	1 1
5	7 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 1 0 0 0 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 1 1 1 1 0 0 0 1 1 1 0 1 0 0

Problem E. Echoes of 24

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

请注意本题的特殊空间限制。



这是立华奏，她非常可爱。

在那寂静的死后世界中，遗忘的记忆仿佛遥远的钟声，回荡在风中。一棵巨大的生命之树，悄然连接起每一颗尚未安息的灵魂的遗憾。

立华奏，那个沉默的守护者，在这片无声的领域中再次醒来。树上的每一个节点都藏着一个数字 a_i ——是情感的碎片、未完成的心愿，以及逐渐淡去的光芒。

为了唤醒那些散落在世界各地的伙伴们，奏必须沿着一条从 l 到 r 的简单路径前行。路径上的值依次为 w_1, w_2, \dots, w_k 。她以第一个节点的值 w_1 作为起点，作为她计数器的初始值。

在接下来的 $k - 1$ 步中，她会：

1. 将下标加 1；
2. 选择将当前值 w_{id} 与计数器 **相加** 或 **相乘**。

她的使命是：将最终的计数器值变成 24。这是一个象征——代表完整的一天、24 小时的陪伴、24 个未说出口的心声。

有些操作会询问某条路径是否可以得到 24；另一些则会更改节点上的记忆值。

立华奏，是否能改写命运，走到那个充满宁静与救赎的结局？

Input

第一行两个整数 n 和 q ($1 \leq n, q \leq 5 \times 10^5$)，表示节点数量与操作数量；

接下来一行 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)，表示每个节点上记录的记忆值；

接下来 $n - 1$ 行，每行两个整数 u, v ，表示一条连接节点 u 和 v 的边；

接下来 q 行，每行为以下两种之一的操作格式：

- 1 l r: 查询从节点 l 到 r 的路径是否可以通过上述操作使计数器变为 24；
- 2 x d: 将第 x 个节点的记忆值更新为 d 。

Output

对于每个 $1 \leq l \leq r$ 的询问，输出一行：

- 若能得到 24，输出 1；
- 否则输出 0。

Examples

standard input	standard output
9 8	1
1 3 1 2 4 6 5 2 2	1
1 2	0
1 3	1
2 4	0
2 5	0
3 6	1
3 7	
6 8	
7 9	
1 4 8	
1 4 9	
1 2 8	
1 6 9	
1 9 6	
2 3 8	
1 6 9	
1 9 6	
8 8	1
3 2 3 3 1 3 1 2	1
2 1	0
3 2	1
4 1	
5 2	
6 4	
7 6	
8 7	
2 4 4	
1 8 2	
2 5 2	
1 2 4	
1 5 2	
2 1 3	
1 4 8	
2 6 2	

Note

- $1 \leq n, q \leq 5 \times 10^5$
- $1 \leq a_i, d \leq 10^9$
- $1 \leq x, l, r \leq n$

Problem F. For the treasury!

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

公元11世纪初，有一些被称为维京人的丹麦入侵者在英格兰活动。

Askeladd是一群维京海盗的领袖，他在这片肥沃的土地上寻找宝藏。在一夜袭击了一个村庄后，他们共收集了 n 个宝藏，按顺序排列，价值分别为 a_1, a_2, \dots, a_n 。Askeladd的团队有一项关于如何分配这些宝藏的协议：作为领袖的Askeladd将取走**前面**的 k 个宝藏，即价值为 a_1, a_2, \dots, a_k 的宝藏，而其余海盗将分配其余的宝藏。但是，由于时间太晚，他们决定在第二天早上进行这个分配。

Askeladd非常聪明且狡猾，他在夜间偷偷摸摸地试图重新安排宝藏，以便他可以获得更多的宝藏总价值。然而，由于某种原因，他只被允许**交换两个相邻的宝藏**。此外，考虑到被其他海盗发现的风险，每次交换将花费Askeladd c 的价值。Askeladd可以执行任意数量（可能为零）的交换。

Askeladd想知道他可以获得的最大利润是多少（即他可以获得的所有宝藏总价值减去交换所消耗的总代价）。

Input

第一行包含三个整数 n, k 和 c ($0 \leq k \leq n \leq 3 \times 10^5, 0 \leq c \leq 10^9$)，表示宝藏的总数量，Askeladd 取走的宝藏数量，以及交换相邻两个宝藏的成本。

接下来一行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)，表示宝藏的价值。

Output

在一行中输出一个整数，表示Askeladd能获得的最大利润。


Examples

standard input	standard output
3 2 1 1 3 5	6
3 2 2 1 3 5	4
4 2 1 2 3 5 6	7
7 3 2 2 1 3 6 10 9 7	10

Note


为了帮助理解，我们提供了第一个样例的图示说明。当Askeladd不进行任何交换时，他获得的总利润等于4。


Times of swaps: 0




Total profit = $(1 + 3) - 0 * 1 = 4$

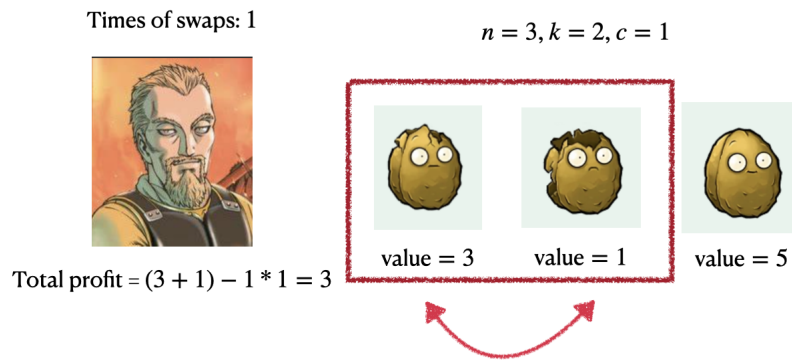
$n = 3, k = 2, c = 1$


value = 1

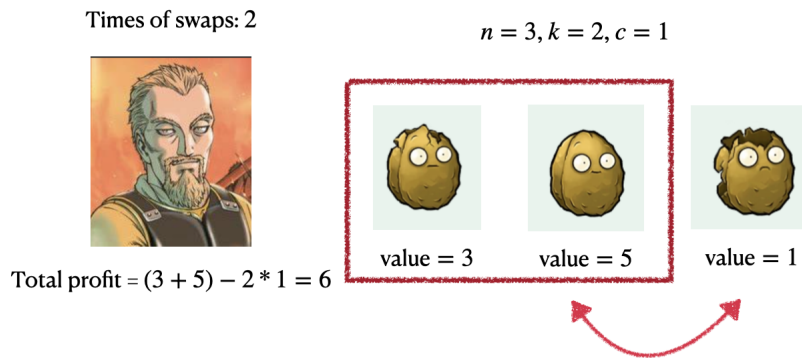

value = 3


value = 5

Askeladd的最佳方法是先交换第一个和第二个宝藏,



然后交换第二和第三个宝藏, 总利润为6。



Problem G. Ghost in the Parentheses

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

爱丽丝有一个有效的括号序列 s^* ，她想通过一个嘈杂的通道与鲍勃分享它。对于每个字符，该通道以 $1/2$ 的概率独立地将该字符传输给鲍勃，或者以 $1/2$ 的概率将一个无法区分的字符 ‘?’ 传输给鲍勃。例如，如果爱丽丝的有效括号序列是 $s = (())$ ，鲍勃可能收到字符序列 $(?)()$ 、 $?(??)?$ ，但不能收到字符序列 $??(())$ 。在收到消息后，鲍勃尝试重建爱丽丝传输的有效括号序列。然而，爱丽丝知道在某些情况下，重建不是唯一的：例如，当收到字符序列 $?(??)?$ 时，有两个可能的重建： $(())$ 和 $((()))$ 。请帮助爱丽丝计算鲍勃收到的消息能够唯一重建有效括号序列的概率。输出答案模 998 244 353。

Input

输入由一行组成，包含一个有效的括号序列 s ($2 \leq |s| \leq 10^6$)。

Output

输出一个整数，表示鲍勃收到的消息能够唯一重建有效括号序列的概率，模 998 244 353。

Examples

standard input	standard output
$()()$	249561089
$((()((()())))$	936828929
$()((()((())))$	826671105

Note

对于第一个示例，只有当第二个字符和第三个字符都没有丢失时，鲍勃收到的消息才能唯一重建有效括号序列。发生此事件的概率为 $\frac{3}{4} \equiv 249\,561\,089 \pmod{998\,244\,353}$ 。

*括号序列 s 是有效的，如果以下任一条件成立：

- s 是空的；
- $s = '(' + t + ')'$ ，其中 t 是一个有效的括号序列；
- $s = t_1 + t_2$ ，其中 t_1 和 t_2 是有效的括号序列。

Problem H. Highs and Lows

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 1024 megabytes

给定整数 a_1, a_2, \dots, a_n 和 M , 请找出一个多项式 $f(x) = \sum_{i=0}^k c_i x^i$, 其系数为整数, 并满足

- 对于所有 $1 \leq i \leq n$, $f(i) \equiv a_i \pm 1 \pmod M$ 。
- k 被最小化。

Input

输入的第一行包含整数 n 和 M ($1 \leq n \leq 30, 2 \leq M \leq 10^9$)。

输入的第二行包含 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i < M$)。

请注意, M 不一定是一个质数。

Output

如果不存在解, 输出 -1 在一行中。

否则, 在一行中输出一个整数 k ($0 \leq k \leq 10^6$)。然后, 输出 $k+1$ 个整数 c_0, c_1, \dots, c_k ($0 \leq c_i < M$), 表示多项式的系数。如果存在多个解, 输出任意一个。

Examples

standard input	standard output
5 998244353 1 3 5 9 11	1 0 2
5 998244353 1 4 9 16 25	2 998244352 0 1
7 19 1 1 2 3 5 8 13	5 13 18 13 1 1 11
6 998244353 1 2 4 8 16 31	4 0 748683264 707089751 249561088 291154603

Problem I. I, Box

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

推箱子 (Sokoban) 是一款由今林博之 (Hiroyuki Imabayashi) 于1981年创作的益智视频游戏系列, 玩家在仓库中推动箱子到指定的存储位置。尽管规则简单, 推箱子展现出复杂而具有挑战性的行为, 吸引了众多益智游戏爱好者和理论计算机科学家。1996年, Dorit Dor和Uri Zwick证明了推箱子是NP完全的, 1997年, Joseph C. Culberson进一步证明了它是PSPACE完全的。至今, 推箱子在玩家中仍然受欢迎, 许多现代益智游戏都受到其机制的启发, 包括*Stephen's Sausage Roll*、*Baba Is You*和*Patrick's Parabox*。



推箱子及一些流行的基于推箱子的益智游戏。

Bobo最近也在玩推箱子, 发现完成一个关卡非常困难。感到沮丧的他抱怨道: "成熟的箱子应该学会自己推动!" 然后, 叮! 魔法发生了。突然, 推动者从推箱子中消失了, 你可以命令箱子自己推动。现在, 游戏变得容易多了! ……或者并非如此?

正式来说, 你需要解决以下没有推动者的推箱子问题:

给定一个被墙壁包围的 $n \times m$ 二维网格, 网格中可能还包含额外的内部墙壁。网格上有几个箱子和目标位置。箱子的数量等于目标位置的数量。保证没有箱子或目标位置占据墙壁单元, 并且所有箱子和所有目标位置都位于不同的位置。

你可以重复执行以下操作:

- 选择任意一个箱子, 并将其向四个基本方向 (上、下、左或右) 中的一个方向移动一格, 前提是目标位置不是墙壁或另一个箱子。

你的任务是确定是否可以将所有箱子移动到目标位置。如果可以, 构造一个这样的移动序列。

Input

输入的第一行包含三个整数 n, m ($1 \leq n, m \leq 50$), 分别表示地图的高度、宽度和箱子的数量。
接下来有 n 行, 每行包含一个长度为 m 的字符串, 表示地图。每个字符属于集合 $\{‘\#’, ‘.’, ‘@’, ‘*’, ‘!’\}$, 其中对于第 i 行的第 j 个字符 ($1 \leq j \leq m$):

- ‘#’字符表示在位置 (i, j) 有一堵墙;
- ‘.’字符表示位置 (i, j) 是空的;
- ‘@’字符表示在位置 (i, j) 有一个箱子;
- ‘*’字符表示在位置 (i, j) 有一个目标位置;
- ‘!’字符表示在位置 (i, j) 同时有一个箱子和一个目标位置。

地图被墙壁包围。输入保证箱子的数量等于目标位置的数量, 并且地图中最多有50个箱子。

Output

如果不存在有效的序列将所有箱子移动到目标位置, 输出 -1 。
否则, 在一行中输出一个整数 ℓ ($0 \leq \ell \leq 10^5$), 表示移动序列的长度。然后, 输出 ℓ 行, 每行包含两个整数和一个字符: $x\ y\ c$ ($1 \leq x \leq n, 1 \leq y \leq m, c \in \{L, R, U, D\}$)。这表示将当前位置为 (x, y) 的箱子向左、右、下或上移动一步, 即移动到 $(x, y-1)$ 、 $(x, y+1)$ 、 $(x+1, y)$ 或 $(x-1, y)$ 。请注意, 你不应该将任何箱子移动到墙壁或另一个箱子上。
如果存在多个有效的移动序列, 输出其中任何一个都将被视为正确。可以证明, 在给定的约束下, 如果存在任何有效的移动序列, 则在 10^5 步内存在这样的序列。

Examples

standard input	standard output
3 3 @.# #.# #.*	4 1 1 R 1 2 D 2 2 D 3 2 R
3 3 #@# @!* ##*	4 2 2 D 1 2 D 2 2 R 2 1 R
3 3 @## #.. **#	-1

Problem J. Journey Around the World

Input file: **standard input**
Output file: **standard output**
Time limit: 8 seconds
Memory limit: 1024 megabytes

有一个 n 行 n 列的网格 ($3 \leq n \leq 200$)。对于 $1 \leq i, j \leq n$ ，在第 i 行第 j 列的单元格与第 i 行第 $(j+1)$ 列的单元格之间有一条无向边（第 $(n+1)$ 列被视为第 1 列），权重为 1 或 2。对于 $1 \leq i < n$ 和 $1 \leq j \leq n$ ，在第 i 行第 j 列的单元格与第 $(i+1)$ 行第 j 列的单元格之间也有一条无向边，权重同样为 1 或 2。

对于 $1 \leq i \leq n$ ，找到从第 1 行第 i 列的单元格到第 n 行第 i 列的单元格的最短路径长度，要求经过每一列至少一次。

Input

第一行包含一个正整数 T ，表示测试用例的数量。所有测试用例中 n 的总和不超过 200。

对于每个测试用例，第一行包含一个正整数 n 。

接下来的 n 行每行包含 n 个整数，其中第 i 行的第 j 个整数表示第 i 行第 j 列的单元格与第 i 行第 $(j+1)$ 列的单元格之间的边的权重。

接下来的 $n-1$ 行每行也包含 n 个整数，其中第 i 行的第 j 个整数表示第 i 行第 j 列的单元格与第 $(i+1)$ 行第 j 列的单元格之间的边的权重。

Output

对于每个测试，输出总共 n 行，其中第 i 行包含一个整数，表示从第 1 行第 i 列的单元格到第 n 行第 i 列的单元格的最短路径长度，要求经过每一列至少一次。

Example

standard input	standard output
2	7
3	7
2 1 2	7
2 1 2	11
2 2 2	11
1 2 2	11
1 1 2	11
5	11
1 1 2 1 1	
2 1 1 2 2	
2 2 1 2 2	
2 1 2 2 1	
2 2 1 1 1	
1 2 2 2 2	
2 1 2 1 2	
2 2 2 1 2	
1 1 2 2 2	

Problem K. Kindred Sums

Input file: `standard input`
Output file: `standard output`
Time limit: 5 seconds
Memory limit: 1024 megabytes

给定 $N = 1000$ 个正整数 a_1, a_2, \dots, a_N 和 $M = 10^{18}$, 请找到两个不同且不重叠的子集 $S, T \subseteq \{1, 2, \dots, N\}$, 使得 $\sum_{i \in S} a_i - \sum_{i \in T} a_i$ 是 M 的倍数。

可以证明, 在给定条件下, 至少存在一个有效的解决方案。

Input

输入由一行组成, 包含 $N = 1000$ 个非负整数 a_1, a_2, \dots, a_N ($1 \leq a_i \leq 10^{18}$)。

Output

输出由一行组成, 该行包含一个长度为 $N = 1000$ 的字符串, 仅包括 0、1 和 2。其中

- 第 i 个位置为 0 表示该元素既不在 S 中, 也不在 T 中;
- 第 i 个位置为 1 表示该元素在 S 中;
- 第 i 个位置为 2 表示该元素在 T 中。

如果有多个有效解, 输出其中任意一个。

请注意, 您的程序应输出一个字符串 (即 012012), 而不是一个整数序列 (即 0 1 2 0 1 2)。

Problem L. Ladder Challenge

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 1024 megabytes

小明和 n 个选手正在参加一个"天梯"游戏，每个选手都有一个积分，选手的积分按严格递增顺序排列，即：

$$a_1 < a_2 < \cdots < a_n$$

小明初始不在这 n 个选手中。现在小明想通过"挑战"来提升自己的排名。
每次挑战，小明会从积分比他高的选手中选择**积分最低**的一个进行挑战。挑战规则如下：

- 小明的积分增加 1；
- 被挑战选手的积分减少 1。

共有 q 次询问。每次询问给出小明的初始积分 x ，以及他想达到的目标排名 y （按当前积分排名，并列视为同名次），你要求出小明**至少需要多少次挑战**才能达到该排名。

Input

第一行包含两个整数 n 和 q ($1 \leq n, q \leq 2 \cdot 10^5$)，表示选手数量和询问次数；
第二行包含 n 个严格递增的整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)，表示每个选手的初始积分；
接下来 q 行，每行两个整数 x 和 y ($0 \leq x \leq 10^9, 1 \leq y \leq n + 1$)，分别表示小明的初始积分和目标排名。

Output

对于每次询问，在一行中输出一个整数，表示小明至少需要多少次挑战才能达到目标排名。

Examples

standard input	standard output
5 3	0
2 5 8 11 20	1
25 1	9
3 4	
0 2	
2 3	0
2 3	1
1 3	2
1 2	
1 1	

Problem M. Monopoly

Input file: standard input
Output file: standard output
Time limit: 12 seconds
Memory limit: 1024 megabytes

博博正在与 N 个其他玩家玩大富翁游戏，他们分别拥有 a_1, a_2, \dots, a_N 个金币，而博博没有金币。现在轮到博博了，为了扭转局面，博博希望使用"均衡卡"道具。这个道具允许博博选择若干名玩家 **不包括他自己**，并将他们的金币数量替换为 **所选玩家金币数量的平均值的下取整**。正式来说，当博博使用均衡卡时，他可以选择一个非空子集 $S \subseteq \{1, 2, \dots, N\}$ ，并令 $T = \left\lfloor \frac{\sum_{i \in S} a_i}{|S|} \right\rfloor$ 。对于所有 $i \in S$ ，玩家 i 的金币数量被设置为 T 。

博博有无限数量的"均衡卡"，并且可以在这一轮中使用任意数量的均衡卡。博博希望通过在这一轮中使用"均衡卡"来最小化其他 N 名玩家可能拥有的最大金币数量。请帮助博博计算这个最小值。

Input

输入包含 T 个测试用例 ($1 \leq T \leq 1000$)。每个测试用例的描述如下。
第一行包含一个正整数 N ($2 \leq N \leq 50$)，表示其他玩家的数量。
接下来的行包含 N 个非负整数 a_1, a_2, \dots, a_N ($0 \leq a_i \leq 50$)，表示其他玩家拥有的金币数量。

Output

对于每个测试用例，输出一行中的一个整数，表示答案。

Example

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