

2025年第一届哈尔滨工业大学程序设计竞赛一 校三区联合校赛

哈尔滨、威海、深圳

2025年12月28日

题号	题目名称
----	------

A	Hello, Harbin Institute of Technology!
---	--

B	Bingo!
---	--------

C	约数游戏
---	------

D	小丑牌
---	-----

E	不是小丑牌
---	-------

F	二维码生成器
---	--------

G	回文串魔法
---	-------

H	追忆
---	----

I	子树 mex
---	--------

J	过程淘汰II
---	--------

K	餐券设计
---	------

L	Jump Jump
---	-----------

M	门不能从这一侧打开
---	-----------

请勿在比赛正式开始前打开题面！

题目 A. Hello, Harbin Institute of Technology!

在编程世界里,“Hello, World!”被认为是很多人写下的第一行程序输出。它因简洁直观而广为流传:只需要把一句话打印到屏幕上,就能验证编译环境、运行流程和基本语法是否正确。自从早期经典教材与示例程序开始使用它以来,“Hello, World!”逐渐成为各类语言、各类平台最常见的入门传统。

今年的哈工大联合秋季校赛也延续了这个传统,不过我们决定做一点小小的变化,考一考你是否知道学校的英文缩写:你需要输出小写的 hello 和小写的哈尔滨工业大学英文缩写,中间用空格符分开。

例如:‘hello hlt’。(请注意这里的英文缩写不一定正确。)

输入

本题无输入。

输出

输出一行内容,见题意描述。

样例

standard input	standard output
(no input)	hello hlt

注释

注意,样例输出仅为格式参考,不一定为正确答案。

题目 B. Bingo!

Bingo 是一种趣味性很强的游戏机制。在一局游戏开始前，需要准备 25 个任务，并将它们放入一个 5×5 的方格表中。游戏过程中，玩家通过完成相应任务来点亮对应的格子；当任意一行、任意一列，或任意一条对角线上的所有格子都被点亮时，即达成 Bingo。通过合理设计任务内容，这种玩法能够灵活应用于多种不同场景，具有较高的通用性和扩展性。

在本题中，你需要完成一局 Bingo 游戏的结算。给定一个 5×5 的矩阵，表示游戏结束时的 Bingo 图。矩阵的每个元素为 0 或 1，其中 1 表示该任务已完成，0 表示未完成。请依据以下规则，判定这场游戏的结果。

1. 游戏胜利：若某一行、某一列、或某一条对角线的 5 个任务全部完成，则判定为游戏胜利。
2. 积分规则：若未胜利，你需要根据如下规则计算积分：
 - (a) 每一个完成的任务都会得到积分，其积分取决于它贡献的最长连线长度：
 - 若该任务与其它完成的任务共同构成的最长连线为 4，则积分为 3；
 - 若该任务与其它完成的任务共同构成的最长连线为 3，则积分为 2；
 - 否则，积分为 1；
 - (b) 最终积分为所有已完成任务的积分之和。

这里，连线指的是横线、竖线或斜线。形式化地，记第 x 行第 y 列的任务为 (x, y) ，则：

- 已完成的任务集合 $\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\} (y_1 \leq y_2 \leq \dots \leq y_k)$ 构成一条长度为 k 的横线，当且仅当 $x_i = x_{i-1}, y_i = y_{i-1} + 1 (2 \leq i \leq k)$ ；
- 已完成的任务集合 $\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\} (x_1 \leq x_2 \leq \dots \leq x_k)$ 构成一条长度为 k 的竖线，当且仅当 $y_i = y_{i-1}, x_i = x_{i-1} + 1 (2 \leq i \leq k)$ ；
- 已完成的任务集合 $\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\} (y_1 \leq y_2 \leq \dots \leq y_k)$ 构成一条长度为 k 的斜线，当且仅当 $x_i = x_{i-1} + 1, y_i = y_{i-1} + 1 (2 \leq i \leq k)$ 或 $x_i = x_{i-1} - 1, y_i = y_{i-1} + 1 (2 \leq i \leq k)$ 。

输入

输入共 5 行，每行包含一个长度为 5 的字符串，构成 5×5 的矩阵，表示任务完成情况。保证矩阵的每个字符 $c \in \{0, 1\}$ 。

输出

若游戏胜利，请输出 “Bingo!”；

否则，输出一个整数，表示玩家的得分。

样例

standard input	standard output
10000 01000 00100 00010 00001	Bingo!
00001 10000 01000 01110 00010	17

注释

对于样例 2，每个格子的积分如下图所示。

				1
3				
	3			
	2	3	2	
			3	

题目 C. 约数游戏

小 A 和小 B 最近迷上了约数游戏，游戏规则如下：

初始有两个正整数 a, b ，双方轮流进行一次修改操作，修改操作有两种选择：

1. 将 a 改为小于它的某个约数 c ，即 $c < a$ 且 $a \bmod c = 0$ 。
2. 将 b 改为小于它的某个约数 d ，即 $d < b$ 且 $b \bmod d = 0$ 。

当一方无法进行修改操作时，另一方获胜。

现在小 A 作为先手，小 B 作为后手。他们想知道，若双方足够聪明，且都采取最优策略进行修改操作，那么先手能否必胜？若先手必胜，则输出 Yes；若先手必败，则输出 No。

输入

本题每个测试数据包含多组测试用例。

第一行，输入一个整数 T ($1 \leq T \leq 10^3$)，表示测试用例组数。

接下来 T 行，每行输入两个整数 a, b ($1 \leq a, b \leq 10^9$)。

输出

输出共 T 行，对于每组测试用例都输出一行 Yes 或 No，表示先手必胜或必败。

样例

标准输入	标准输出
6	No
1 1	Yes
4 1	Yes
4 3	No
4 9	No
6 9	Yes
8 9	

注释

样例解释如下：

第一组测试用例， $a = 1, b = 1$ ，初始先手无法进行修改操作。

因此先手必败，输出 No。

第二组测试用例， $a = 4, b = 1$ ，初始先手将 a 改为 1，后手便无法进行修改操作。

因此先手必胜，输出 Yes。

第三组测试用例， $a = 4, b = 3$ ，初始先手将 a 改为 2，变成 $a = 2, b = 3$ 局面，此后：

- 若后手将 a 改为 1，则先手将 b 改为 1，后手便无法进行修改操作，先手必胜。
- 若后手将 b 改为 1，则先手将 a 改为 1，后手便无法进行修改操作，先手必胜。

因此先手必胜，输出 Yes。

第四组测试用例， $a = 4, b = 9$ ，初始：

- 若先手将 a 改为 1，则后手将 b 改为 1，先手便无法进行修改操作，先手必败。
- 若先手将 b 改为 1，则后手将 a 改为 1，先手便无法进行修改操作，先手必败。
- 若先手将 a 改为 2，则后手将 b 改为 3，此时变成 $a = 2, b = 3$ 局面，先手必败。
- 若先手将 b 改为 3，则后手将 a 改为 2，此时变成 $a = 2, b = 3$ 局面，先手必败。

因此先手必败，输出 No。

第五组测试用例， $a = 6, b = 9$ ，初始：

- 若先手将 a 改为 1，则后手将 b 改为 1，先手便无法进行修改操作，先手必败。
- 若先手将 b 改为 1，则后手将 a 改为 1，先手便无法进行修改操作，先手必败。
- 若先手将 a 改为 2，则后手将 b 改为 3，此时变成 $a = 2, b = 3$ 局面，先手必败。
- 若先手将 a 改为 3，则后手将 b 改为 3，此时变成 $a = 3, b = 3$ 局面，先手必败。
- 若先手将 b 改为 3，则后手将 a 改为 2，此时变成 $a = 2, b = 3$ 局面，先手必败。

因此先手必败，输出 No。

第六组测试用例， $a = 8, b = 9$ ，初始先手将 a 改为 4，此时变成 $a = 4, b = 9$ 局面。

因此先手必胜，输出 Yes。

题目 D. 小丑牌

你正在进行一场小丑牌对局。当前回合，你手中有 N 张手牌和 M 张备选的小丑牌。你需要通过"出牌"和"小丑牌调度"两个阶段的决策，来获得本回合的最高得分。

1. 卡牌与牌面筹码

本游戏剔除了 K, Q, J 三种人头牌，只保留数字牌和 A。每张牌由点数和花色组成（例如 Th 代表红桃 10，9c 代表梅花 9）。每张牌根据其点数拥有固定的牌面筹码，规则如下：

- 2, 3, ..., 9: 牌面筹码等于点数。
- T (10): 牌面筹码为 10。
- A (Ace): 牌面筹码为 11。

2. 出牌规则

你需要从 N 张手牌中，严格选取 K 张 ($1 \leq K \leq 5$) 组成下表定义的一种合法牌型打出。

- 选出的 K 张牌必须完全符合牌型的定义（张数必须精确匹配）。
- 每种牌型对应一个基础筹码 (X) 和基础倍率 (Y)。

注意：手牌中可能出现相同的牌

牌型名称	所需张数	组成严格定义	基础筹码 (X)	基础倍率 (Y)
高牌 (High Card)	1	任意一张单牌	5	1
对子 (Pair)	2	两张点数相同的牌	10	2
三条 (Three of a Kind)	3	三张点数相同的牌	30	3
四条 (Four of a Kind)	4	四张点数相同的牌	60	7
同花 (Flush)	5	五张花色相同的牌	35	4
葫芦 (Full House)	5	3张点数相同 + 2张点数相同	40	4

注意：若手牌组合同时符合多种牌型定义，你可以选择任意一种符合定义的牌型进行结算。

3. 小丑牌 (Joker) 机制

你拥有 M 张备选小丑牌，每张牌拥有一个效果类型和数值 V ：

- 类型 1: 使当前总筹码增加 V 。
- 类型 2: 使当前总倍率增加 V 。
- 类型 3: 使当前总倍率乘以 V 。

你需要从 M 张中挑选至多 5 张小丑牌生效。为了最大化最终得分，你可以任意安排这 5 张牌的生效顺序。

4. 得分计算流程

一次出牌的最终得分严格按照以下步骤计算：

1. 初始统计:

- 设初始总筹码 $C = \text{基础筹码 } X + \sum(\text{所选出的手牌的牌面筹码})$ 。
- 设初始总倍率 $M = \text{基础倍率 } Y$ 。

2. 小丑结算: 按照你设定的顺序, 依次将所选小丑牌的效果应用到 C 或 M 上。

- 若应用类型 1, 则 $C \leftarrow C + V$ 。
- 若应用类型 2, 则 $M \leftarrow M + V$ 。
- 若应用类型 3, 则 $M \leftarrow M \times V$ 。

3. 最终得分: $Score = C \times M$ 。

输入

第一行包含一个整数 T ($1 \leq T \leq 10$), 表示测试数据组数。

对于每组数据: 第一行包含两个整数 N, M ($1 \leq N \leq 15, 0 \leq M \leq 15$), 分别表示手牌数量和小丑牌数量。

第二行包含 N 个字符串, 表示手牌。格式为"点数+花色" (例如 Th、9c、Ad)。

- 点数字符: 2, 3, ..., 9, T, A。
- 花色字符: h (红桃), d (方片), s (黑桃), c (梅花)。

接下来 M 行, 每行描述一张小丑牌, 包含两个整数 $Type$ 和 Val 。

- $Type \in \{1, 2, 3\}$ 分别代表: (1) 增加筹码, (2) 增加倍率, (3) 乘倍率。
- 对于 $Type = 1$, $1 \leq Val \leq 500$ 。
- 对于 $Type = 2$, $1 \leq Val \leq 50$ 。
- 对于 $Type = 3$, $1 \leq Val \leq 10$ 。

输出

对于每组数据, 输出一行一个整数, 表示在最优策略下能获得的最高得分。

样例

standard input	standard output
1 5 3 Ah Ad As Ac 9h 1 100 2 20 3 2	11016

注释

样例 1 解释:

假设手牌为: Ah Ad As Ac 9h (四张 A 和一张 9)。最优策略是打出 四条 (Ah, Ad, As, Ac)。

- 查表得基础值: 基础筹码 60, 基础倍率 7。
- 手牌牌面筹码和: $11 + 11 + 11 + 11 = 44$ 。
- 此时初始状态: 总筹码 $C = 60 + 44 = 104$, 总倍率 $M = 7$ 。

假设你选择的小丑牌为: (类型1, 数值100), (类型2, 数值20), (类型3, 数值2)。为了最大化分数, 最优的生效顺序如下:

1. 应用类型 1 (+100 筹码): C 变为 $104 + 100 = 204$ 。
2. 应用类型 2 (+20 倍率): M 变为 $7 + 20 = 27$ 。
3. 应用类型 3 ($\times 2$ 倍率): M 变为 $27 \times 2 = 54$ 。

最终得分: $204 \times 54 = 11016$ 。

题目 E. 不是小丑牌

Alice 最近迷上了一款卡牌游戏，但她玩的规则有点特别。牌堆中只有三种花色的牌：梅花 (Clubs)、黑桃 (Spades) 和 方块 (Diamonds)。初始时，Alice 手中分别有 C 张梅花、 S 张黑桃和 D 张方块。

游戏规则如下：Alice 每次操作必须从 当前数量最多 的两种花色中各取出一张，将其消耗掉，并合成一张 第三种花色 的牌。例如，如果当前梅花和黑桃的数量最多，她就消耗一张梅花和一张黑桃，获得一张方块。

Alice 会一直进行这个操作，直到无法进行为止（即剩下的花色种类少于2种）。现在她想知道，如果她严格遵守上述规则（若有多种花色数量并列最多，则存在多种选择方案），最终剩下的那张牌的花色是否是唯一确定 的？如果是，请输出剩下的那张牌的花色；否则（如果最终无法只剩下一张牌，或者存在多种可能的最终花色），请输出 NO。

输入

第一行包含一个正整数 T ($1 \leq T \leq 10$)，表示测试数据组数。接下来 T 行，每行包含三个整数 C, S, D ($1 \leq C, S, D \leq 8$)，分别表示梅花、黑桃和方块的初始数量。

输出

对于每组数据，如果最终剩下的牌花色唯一且只剩一张，输出对应的花色英文单词：

- 如果是梅花，输出 Clubs
- 如果是黑桃，输出 Spades
- 如果是方块，输出 Diamonds

否则，输出 NO。

样例

standard input	standard output
2	Spades
1 2 3	NO
2 2 2	

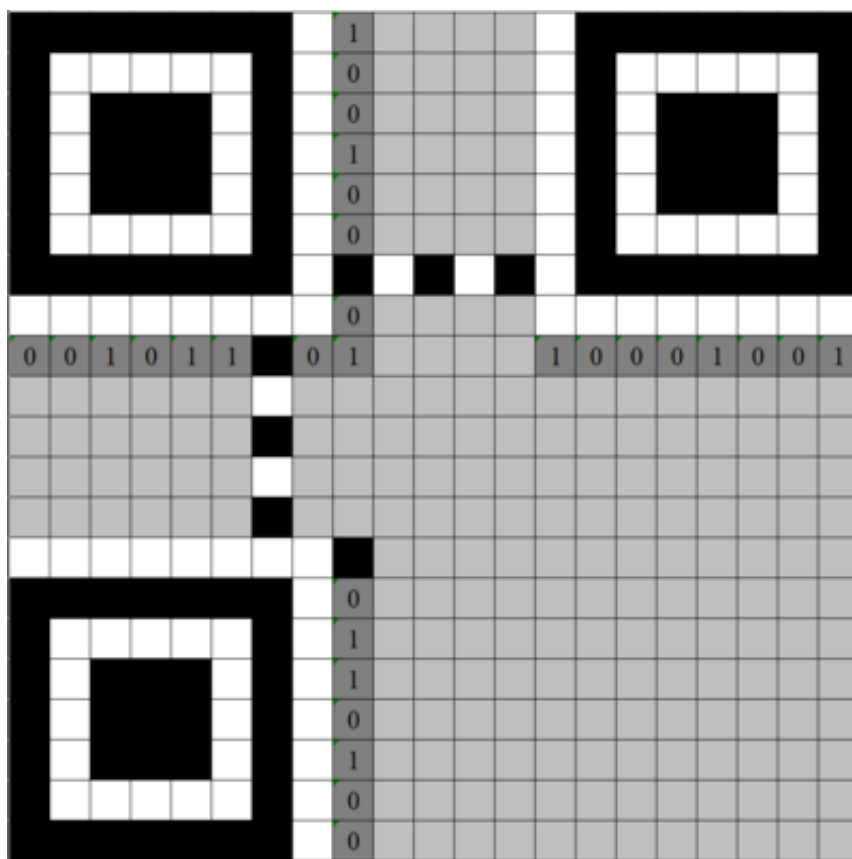
注释

在第一组样例 (1, 2, 3) 中：1. 方块(3)和黑桃(2)最多，消耗一张方块和黑桃，生成梅花。当前变为：(2, 1, 2)。2. 梅花(2)和方块(2)最多，消耗一张梅花和方块，生成黑桃。当前变为：(1, 2, 1)。3. 黑桃(2)和梅花(1)最多（或黑桃和方块），无论选哪个，结果对称。假设选黑桃和梅花，生成方块。当前变为：(0, 1, 2)。4. 方块(2)和黑桃(1)最多，消耗它们，生成梅花。当前变为：(1, 0, 1)。5. 梅花(1)和方块(1)最多，消耗它们，生成黑桃。当前变为：(0, 1, 0)。最终只剩下一张黑桃，虽然过程中可能有多种选择方法（如第3步），最终结果相同，故输出 Spades。

题目 F. 二维码生成器

QR Code 是日常生活中常用的一种二维码，可视为记录数据信息的黑白像素矩阵，信息的形式为 01 比特串，黑色方格代表 1，白色方格代表 0。QR Code 现有 40 个版本，版本 1 的大小为 21×21 ，版本 40 的大小为 177×177 ，越大则可容纳的信息量越多。QR Code 提供数字模式，字母数字模式，字节模式等多种模式来记录数据信息。QR Code 还具备纠错功能，有 L、M、Q、H 这 4 个纠错级别，级别越高纠错能力越强，但能容纳的数据量就越少。

本题研究 QR Code 最简单的结构：QR Code 版本 1，采用数字模式，H 纠错级别，其结构如下图所示：



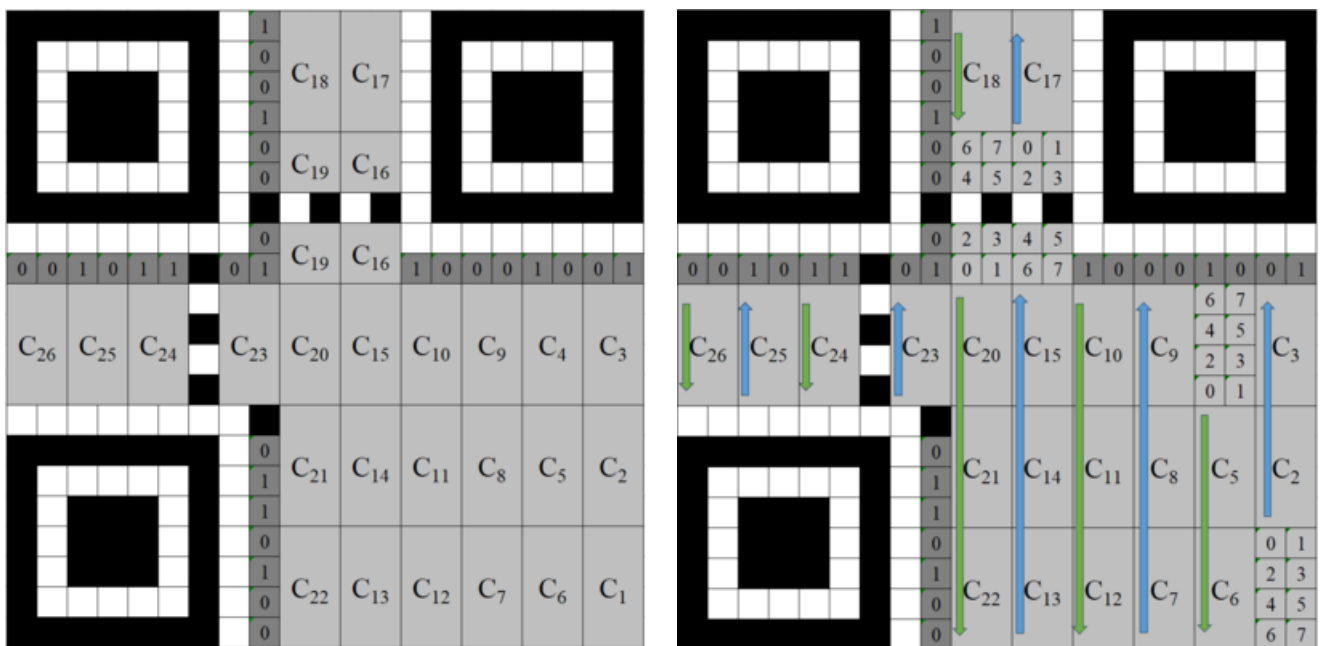
该结构是一个 21×21 的 01 矩阵，左上角坐标为 (0,0)，左下角坐标为 (20,0)，右上角坐标为 (0,20)，右下角坐标为 (20,20)。其中黑色方格表示值固定为 1；白色方格表示值固定为 0；深灰色方格表示的值与模式和纠错级别有关，本题采用数字模式，H 纠错级别，表示的值和上图所示一致，即上图中深灰色方格填 1 则表示值为 1，填 0 则表示值为 0；浅灰色方格表示的值与存储的数据信息有关，需要通过以下几个步骤确定：

1. 编码数据码字
2. 生成纠错码字
3. 码字比特流绘制
4. 掩膜图案绘制

其中，第二步生成纠错码字需要计算生成多项式的系数，并根据生成多项式计算出余数多项式的系数。该过程较为复杂，因此本题省略前两个步骤，输入数据将直接给出第二步完成后的数据码字和纠错码字，

总共为 26 个码字 $c_1, c_2, \dots, c_{26}$ ($0 \leq c_i \leq 255$)，你需要根据这 26 个码字的值实现后两个步骤，并输出最终的二维码结构。

第三步为码字比特流绘制。如下图所示，浅灰色方格总共有 208 个，从右下角开始按 S 形划分为 C_1 到 C_{26} 共 26 个区域，每个区域占 4 行 2 列共 8 个方格，区域 C_i 所填的值由码字 c_i 决定。每个区域内的格子编号为 0 ~ 7，若区域编号方向由下至上（图中蓝色箭头所示），则区域内左上角格子编号为 0，右下角格子编号为 7，如区域 C_1, C_{16} 所示；若区域编号方向由上至下（图中绿色箭头所示），则区域内左上角格子编号为 0，右上角格子编号为 7，如区域 C_4, C_{19} 所示。区域中编号 0 ~ 7 的格子所填的值为将 c_i 转换成 8 位二进制数后第 0 ~ 7 位上的值。例如，数据信息 114514 经第二步完成后 $c_3 = 114$ ，转换为 8 位二进制数即为 01110010，其中第 1, 4, 5, 6 位为 1，第 0, 2, 3, 7 位为 0，因此区域 C_3 编号为 1, 4, 5, 6 的格子填 1，编号为 0, 2, 3, 7 的格子填 0。



第四步为掩膜图案绘制。本题采用的掩膜图案表达式为 $(i + j) \bmod 2 = 0$ ，即对于坐标为 (i, j) 的浅灰色方格，第三步的所填值与 $[(i + j) \bmod 2 = 0]$ 异或后的结果为最终所填值。其中 $[P]$ 表示当 P 为真时结果为 1，当 P 为假时结果为 0。请注意，只有浅灰色方格进行掩膜操作，而黑色、白色和深灰色方格都不进行掩膜操作。

现在请编写一个程序作为二维码生成器，输入数据码字和纠错码字，输出对应的 21×21 二维码，请用字符 # 表示值为 1 的方格，字符 . 表示值为 0 的方格。

输入

输入共三行，26 个整数，表示码字 $c_1, c_2, \dots, c_{26}$ ($0 \leq c_i \leq 255$)。

第一行，输入 10 个整数，表示前 10 个码字 $c_1, c_2, \dots, c_{10}$ 。

第二行，输入 10 个整数，表示中间 10 个码字 $c_{11}, c_{12}, \dots, c_{20}$ 。

第三行，输入 6 个整数，表示后 6 个码字 $c_{21}, c_{22}, \dots, c_{26}$ 。

输出

输出共 21 行，每行为 21 个字符 # 或 . 构成的字符串，表示 21×21 的二维码。

样例

标准输入	标准输出
16 24 114 128 128 236 17 236 17 72 228 6 59 233 77 169 40 48 215 80 206 191 120 191 244 59	#####.#.#.##### #....#...###.#....# #.###.#...#...#...# #.###.#...#...#...# #.###.#...#...#...# #.###.#...#...#...# #....#...###.#....# #####.#.#.#######..... ..#...###...#...#...# .#.#...#...##### .#...###...###...#.#.# ..#.#...###...###...# .#.#...###...###...# ..#.#...###...###...##.#...###...### #####...#...#...#... #....#.#...###...###... #.###.#...#...###...# #.###.#...###...###... #.###.#...#...#...# #....#...#...#...#... #####.#...#...#...#

注释

输入样例为数据信息 114514 所对应的码字，若将输出样例中的字符 # 换成黑色方格，字符 . 换成白色方格，得到的 21×21 二维码如下图所示（赛后大家可以用手机微信上的"扫一扫"等扫码软件扫描，均可以扫出 114514）：



题目 G. 回文串魔法

给定一个长度为 n 的字符串 s ,只包含小写字母。

再给定一个长度为 26 的数组 v , 表示修改操作的代价。

你需要对字符串执行如下流程（按顺序）:

- 1.选择两个位置 i,j （允许 $i = j$ ），交换 s_i 与 s_j , 代价为 0。（必须做该操作恰好一次）
- 2.然后你可以对任意位置进行若干次字符修改（也可以不修改）。每次把某位置改为字母 c 的代价为 v_c 。

你的目标是使最终字符串变为回文串，并使总代价最小。

一个长度为 n 的字符串 t 称为回文串，当且仅当对所有 $1 \leq i \leq n$ 都满足 $t_i = t_{n-i+1}$

输入

第一行一个整数 n ($1 \leq n \leq 10^6$)。

第二行一个长度为 n 的仅含小写字母的字符串 s 。

第三行 26 个整数 $v_a,v_b,\dots,v_z$, 其中 v_c ($0 \leq v_c \leq 10^9$) 表示把某个字符改为字母 c 的代价。

输出

输出一行一个整数，表示把 s 变成回文串的最小总代价。

样例

standard input
10 btpyjyyptf 14 18 13 13 12 7 6 4 16 16 6 4 19 11 3 15 14 10 6 10 2 6 14 8 11 7
standard output
4

standard input
6 tarjen 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
standard output
163

题目 H. 追忆

若花胜去年红，坞中莲蕊竟已开已落，醉倒
芳丛，一眼岁月都无穷。

— 东风志

年少时的时光，总是与许多独特的景象相伴：或许是那一缕东风，或许是那一片柳絮，又或许是那一抹剑光、一盏浊酒、一地月光。这些景象早已远去，化作一行行诗句。

你翻阅这卷诗，仿佛又重新走过了那些岁月。诗卷共有 n 行，第 i 行诗描述了第 i 个景象 a_i ，相同的景象可能出现在不同的行中。你打算从中选取连续的一段区间 $[l, r]$ ，作为你最深刻的"忆境"。你认为，一段忆境的价值，不仅在于其中包含了多少种独特的景象，也在于其之外保留了多少种独特的景象。换言之，忆境的价值等于这段诗中不同景象的种类数加上这段诗外不同景象的种类数。

你想知道，忆境的可能的最大价值是多少？

简要题意

给出一个长度为 n 的序列 a ，你需要找到一个区间 $[l, r]$ ，使得该区间中不同的数个数加上该区间外不同的数个数之和最大。请给出这个最大值。

输入

本题含有多组测试数据。

第一行输入一个整数 T ，表示测试数据组数。

对于每组测试数据：

第一行输入一个整数 n ，表示序列 a 的长度。

接下来第二行包含 n 个整数，第 i 个整数表示 a_i 。

输出

对于每组测试数据：

一行包含一个整数表示答案。

样例

standard input	standard output
2	4
4	4
1 2 2 1	
4	
1 2 1 2	

注释

对于 100% 的数据， $1 \leq T \leq 10^6, 1 \leq n \leq 10^6, \sum n \leq 10^6, 1 \leq a_i \leq n$ 。

题目 I. 子树 mex

给定一棵 n 个点的树，1 号点为根节点， i 号点的点权为 a_i 。

对 $i = 1, 2, \dots, n$ 都求：以 i 号点为根的子树内所有节点点权之集的 mex。

其中 $\text{mex } S$ 的定义为可重集 S 中未出现的最小非负整数，例如 $\text{mex}\{0, 1, 0, 3\} = 2$ ， $\text{mex}\{1, 2, 3, 5\} = 0$ 。

输入

输入共三行。

第一行，输入一个整数 n ($1 \leq n \leq 5 \times 10^5$)，表示树的节点总数。

第二行，输入 $n - 1$ 个整数 f_i ($2 \leq i \leq n$, $1 \leq f_i < i$)，表示 i 号点的父节点为 f_i 。

第三行，输入 n 个整数 $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$)，表示节点点权。

输出

输出共一行， n 个以空格相隔的整数，表示 $1 \sim n$ 号点的子树内点权之集的 mex。

样例

标准输入	标准输出
8 1 1 1 2 2 3 7 1 2 3 5 0 1 0 2	4 3 1 0 1 0 1 0

注释

样例解释如下：

- 1 号点的子树内点权之集为 $S_1 = \{1, 2, 3, 5, 0, 1, 0, 2\}$ ， $\text{mex } S_1 = 4$ 。
- 2 号点的子树内点权之集为 $S_2 = \{2, 0, 1\}$ ， $\text{mex } S_2 = 3$ 。
- 3 号点的子树内点权之集为 $S_3 = \{3, 0, 2\}$ ， $\text{mex } S_3 = 1$ 。
- 4 号点的子树内点权之集为 $S_4 = \{5\}$ ， $\text{mex } S_4 = 0$ 。
- 5 号点的子树内点权之集为 $S_5 = \{0\}$ ， $\text{mex } S_5 = 1$ 。
- 6 号点的子树内点权之集为 $S_6 = \{1\}$ ， $\text{mex } S_6 = 0$ 。
- 7 号点的子树内点权之集为 $S_7 = \{0, 2\}$ ， $\text{mex } S_7 = 1$ 。
- 8 号点的子树内点权之集为 $S_8 = \{2\}$ ， $\text{mex } S_8 = 0$ 。

因此输出为 4 3 1 0 1 0 1 0。

题目 J. 过程淘汰II

在深圳的 T3 教学楼，有一个空间异常：这里没有 3 楼。无论是乘坐电梯还是步行楼梯，空间都会直接从 2 楼跳跃到 4 楼。

据说，这是由于 T3 的空间场域极度不稳定，连接任意两个位置的通路，都会被路径上最薄弱的"瓶颈"所主导。这一现象也被称为"过程淘汰"。

为了研究这一现象，你记录了 T3 走廊上 N 个观测点的能量值序列 A ，并试图计算在不同维度下的"总淘汰代价"。

我们发现空间的不稳定性主要源于物质的基本构成。因此，我们将针对 K 的每一个质因子 (Prime Factor) p ，分别计算该维度下的淘汰代价，并将所有结果累加。

对于 K 的每一个质因子 p ，定义当前维度的投影序列 B ，其中 $B_i = A_i \bmod p$ 。在序列 B 上，对于任意两个位置 i 和 j ($1 \leq i < j \leq N$)，它们之间会产生一个淘汰代价 $Cost_p(i, j)$ ，计算规则如下：

1. 确定淘汰点：考察闭区间 $[i, j]$ 内的所有数值，找到值最小的位置。根据 T3 的"先序优先"法则，如果存在多个位置的值得于最小值，我们取下标最小（最左侧）的那个位置作为"淘汰点"，记为 pos 。
2. 计算代价：位置 i 和 j 之间的淘汰代价由两端的能量差异（异或和）与淘汰点的能量共同决定：

$$Cost_p(i, j) = (B_i \oplus B_j) \times B_{pos}$$

形式化地，你需要求解的答案 Ans 定义如下：

$$Ans = \left(\sum_{p \in \mathbb{P}, p|K} \sum_{1 \leq i < j \leq N} ((A_i \bmod p) \oplus (A_j \bmod p)) \times \min_{t=i}^j (A_t \bmod p) \right) \bmod 998244353$$

其中：

- $p \in \mathbb{P}, p|K$ 表示 p 是 K 的质因子（例如：若 $K = 12$ ，则质因子为 2 和 3）。
- $\oplus$ 表示按位异或运算 (Bitwise XOR)。
- $\min_{t=i}^j(\dots)$ 表示取区间 $[l, r]$ 内的最小值（若有多个最小值，取下标最小的对应值）。
- $[l, r] \subseteq \mathbb{Z}$ 表示所有满足 $l \leq x \leq r$ 的整数 x 构成的闭区间（例如：若 $[1, 4]$ ，则区间包含整数 1, 2, 3, 4）。

输入

第一行输入一个整数 t ($1 \leq t \leq 1000$)，表示测试数据组数。

对于每组测试数据：第一行输入两个整数 N 和 K ($1 \leq N \leq 5 \times 10^5, 2 \leq K \leq 10^8$)，分别表示序列长度和场域参数。第二行输入 N 个整数 $A_1, A_2, \dots, A_N$ ($1 \leq A_i \leq 10^9$)，表示观测点的能量值。

保证所有测试数据的 N 之和不超过 5×10^5 。

输出

对于每组测试数据，输出一行一个整数，表示总淘汰代价对 998244353 取模后的结果。

样例

standard input	standard output
3	6
5 6	4
2 5 1 6 3	0
6 10	
2 2 5 3 1 1	
1 6	
9	

注释

对于第一组样例， $N = 5, K = 6$ ，质因子为 2 和 3。

- 当 $p = 2$ 时，投影序列 $B = \{0, 1, 1, 0, 1\}$ 。经计算，所有区间的淘汰代价均为 0。
- 当 $p = 3$ 时，投影序列 $B = \{2, 2, 1, 0, 0\}$ 。其中：
 - 区间 $[1, 3]$ (对应值 $2, 2, 1$) 的最小值为 1 (下标 3)，代价为 $(2 \oplus 1) \times 1 = 3$ 。
 - 区间 $[2, 3]$ (对应值 $2, 1$) 的最小值为 1 (下标 3)，代价为 $(2 \oplus 1) \times 1 = 3$ 。
 - 其他区间代价均为 0。

该维度总代价为 6。

最终答案为 $0 + 6 = 6$ 。

对于第二组样例， $N = 6, K = 10$ ，质因子为 2 和 5。

- 当 $p = 2$ 时，总代价为 0。
- 当 $p = 5$ 时，投影序列 $B = \{2, 2, 0, 3, 1, 1\}$ 。非零贡献来自：
 - 区间 $[4, 5]$ (值 $3, 1$)，最小值为 1，代价 $(3 \oplus 1) \times 1 = 2$ 。
 - 区间 $[4, 6]$ (值 $3, 1, 1$)，最小值为 1 (下标 5)，代价 $(3 \oplus 1) \times 1 = 2$ 。

该维度总代价为 4。

最终答案为 $0 + 4 = 4$ 。

题目 K. 餐券设计

XCPC 的比赛主办方通常会为选手提供餐券，假设主办方计划为每位选手提供 k 张餐券，总面值 n 元。

主办方希望尽量避免找钱或多花钱的情况，因此这 k 张餐券能够恰好表示 $1 \sim n$ 的所有钱数。

同时主办方希望餐券纸张的印制数越少越好，因此 k 需要尽可能地小。

显然对于给定的总面值 n ，餐券张数 k 有最小值 $k_{\min}$ ，当 $k = k_{\min}$ 时可能有多种设计方案，你只需要为主办方提供餐券面值按从小到大排序后的字典序最小和最大的方案。

更准确地说，给定 n ，从满足下列条件的序列 $a_1, a_2, \dots, a_k$ 中：

1. $a_1 + a_2 + \dots + a_k = n$
2. $1 \leq a_1 \leq a_2 \leq \dots \leq a_k \leq n$
3. 对 $m = 1, 2, \dots, n$ ，可重集 $\{a_1, a_2, \dots, a_k\}$ 都存在元素和为 m 的子集。

你需要找到 k 最小的序列，输出 k 的最小值 $k_{\min}$ 以及 $k = k_{\min}$ 时字典序最小和最大的序列。

输入

本题每个测试数据包含多组测试用例。

第一行，输入一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例组数。

接下来 T 行，每行输入一个整数 n ($1 \leq n \leq 10^9$)，表示总面值 n 元。

输出

输出共 $3T$ 行，对于每组测试用例，输出三行：

- 第一行，输出一个整数，表示 k 的最小值 $k_{\min}$ 。
- 第二行，输出 $k_{\min}$ 个以空格相隔的整数，表示序列 $a_1, a_2, \dots, a_{k_{\min}}$ 字典序最小的方案。
- 第三行，输出 $k_{\min}$ 个以空格相隔的整数，表示序列 $a_1, a_2, \dots, a_{k_{\min}}$ 字典序最大的方案。

样例

标准输入	标准输出
2	2
3	1 2
5	1 2
	3
	1 1 3
	1 2 2

注释

样例解释如下：

第一组测试用例， $n = 3$ ：

- $k = 1$ 时，满足前两个条件的序列方案只有 $\{3\}$ ，不存在元素和为 $m = 1, 2$ 的子集。
- $k = 2$ 时，满足前两个条件的序列方案只有 $\{1, 2\}$ ，元素和为 $m = 1, 2, 3$ 的子集都存在。

因此 $n = 3$ 时 $k_{\min} = 2$ ，字典序最小的方案为 $\{1, 2\}$ ，字典序最大的方案为 $\{1, 2\}$ 。

第二组测试用例， $n = 5$ ：

- $k = 1$ 时，满足前两个条件的序列方案只有 $\{5\}$ ，不存在元素和为 $m = 1, 2, 3, 4$ 的子集。
- $k = 2$ 时，满足前两个条件的序列方案有 $\{1, 4\}, \{2, 3\}$ ，其中 $\{1, 4\}$ 不存在元素和为 $m = 2, 3$ 的子集， $\{2, 3\}$ 不存在元素和为 $m = 1, 4$ 的子集。
- $k = 3$ 时，满足前两个条件的序列方案有 $\{1, 1, 3\}, \{1, 2, 2\}$ ，这两个方案的元素和为 $m = 1, 2, 3, 4, 5$ 的子集都存在。

因此 $n = 5$ 时 $k_{\min} = 3$ ，字典序最小的方案为 $\{1, 1, 3\}$ ，字典序最大的方案为 $\{1, 2, 2\}$ 。

题目 L. Jump Jump

跳一跳 (Jump Jump) 是一款风靡一时的数字跳跃游戏。在游戏中, 玩家面对一个由数字 1 到 n 排列而成的台阶序列, 每次可以跳到任意一个比当前数字更大的数字台阶上。游戏的目标是从最左边开始, 通过一系列跳跃到达最右边, 并尽可能地获得更长的连续递增跳跃序列。

最近, 游戏开发者小 G 决定为玩家设计一些新的挑战关卡。他发现, 一个关卡的难度与其中"最长上升子序列" (LIS) 的数量密切相关: LIS 数量越多, 玩家解锁隐藏成就的路径就越多, 关卡也越具有挑战性和趣味性。小 G 很快制作出了 K 较小时的若干关卡, 但他发现, 当 K 很大时, 他并不能想到如何快速制作关卡。

于是, 小 G 求助于你, 因此, 设计新关卡的任务就交给你了: 对于给定的目标数量 K , 你需要构造一个台阶序列 (即一个排列), 使得这个排列中恰好包含 K 个不同的最长上升子序列。注意, 作为关卡设计师, 你可以在保证 $1 \leq n \leq 1500$ 的前提下, 自由选择排列的长度 n 。

输入

第一行输入一个整数 K , 保证 $1 \leq K < 2^{30}$ 。

输出

第一行输出一个正整数 n , 需要保证 $1 \leq n \leq 1500$ 。

第二行输出一个 1 到 n 的排列 p , 你需要保证它的最长上升子序列个数恰好为 K 。

可以证明在题目给出的限制条件下, 总是存在至少一个满足要求的排列。

样例

standard input	standard output
1	3 1 2 3
2	4 3 4 1 2
4	4 2 1 4 3

题目 M. 门不能从这一侧打开

在小 C 的世界里，有 n 个藏宝点， m 条连接藏宝点的双向道路，每条道路都有其特定的长度 l 。

这个世界受到了神奇的诅咒，阻止所有贪婪的寻宝人：有 k 个限制 (x, y, z) 表示从藏宝点 x 走到藏宝点 y 后不能往藏宝点 z 走。注意三元组是有序的，如可以从藏宝点 y 走到藏宝点 x 再走到藏宝点 z 。

形式化地，如果小 C 从藏宝点 1 到藏宝点 T 所经过的藏宝点序列为 $(u_0 = 1, u_1, \dots, u_p = T)$ ，则该序列的任意一个长度为 3 的连续子序列 (u_{i-1}, u_i, u_{i+1}) 不能跟 k 个限制中的任意一个表示的三元组 (x_j, y_j, z_j) 完全相同。两个三元组 (a, b, c) 和 (x, y, z) 完全相同当且仅当 $a = x$ 且 $b = y$ 且 $c = z$ 。

但是小 C 不仅贪婪而且有人脉，他想请你帮他分别计算出从藏宝点 1 出发，到达藏宝点 $1, 2, \dots, n$ 所需的最短路径长度。

输入

第一行输入三个正整数 n, m, k ($1 \leq n, m, k \leq 5 \times 10^5$)，分别表示藏宝点个数，道路个数，限制个数。

接下来 m 行，每行输入三个正整数 u, v, l ($1 \leq u, v \leq n, 1 \leq l \leq 10^9$)，表示有一条连接 u 和 v 长度为 l 的道路。不保证 $u \neq v$ ，但保证对于任意第 i 行和第 j 行 ($i \neq j$) 输入的边 (u_i, v_i) 和 (u_j, v_j) ，不存在 $(u_i = u_j \wedge v_i = v_j)$ 或 $(u_i = v_j \wedge v_i = u_j)$ ，即任意两点之间至多有一条直接相连的道路。

接下来 k 行，每行输入三个正整数 x, y, z ($1 \leq x, y, z \leq n$)，表示一个限制。

输出

输出 n 行。

第 i 行输出一个正整数或 -1 。如果能从藏宝点 1 走到藏宝点 i ，则输出最短路径长度；如果无法到达，则输出 -1 。

样例

standard input	standard output
3 2 1	0
1 2 114514	114514
2 3 1919810	-1
1 2 3	