

2026 年国际大学生程序设计竞赛 全国邀请赛（陕西）

正式赛

2026 年 5 月 2 日



西北工业大学

NORTHWESTERN POLYTECHNICAL UNIVERSITY

题目列表

A	North and South
B	Operating Robot
C	Palindromic and Balanced
D	Generals
E	Registration
F	Split Sticks
G	Transform
H	Unreachable Land
I	VIP Coupon
J	Would You Make a Convex?
K	XOR and LCA
L	Yesterday Once More (Easy Version)
M	Yesterday Once More (Hard Version)
N	Zebra Crossing

本试题册共 14 题，27 页。

如果您的试题册缺少页面，请立即通知志愿者。

主办方



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

命题方

Cfz 出题组

Cfz Problem Setting Group



Problem A. North and South

Input file: standard input
Output file: standard output

Yuki 有一个长度为 n 的序列 a 。

Yuki 定义一次操作为：

- 选择一个 **长度为偶数** 的区间 $[l, r]$ 。对于每个满足 $l \leq i \leq r$ 的正整数 i ：
 - 若 $i - l$ 为奇数，则 a_i 的值减少 1，即 $a_i \leftarrow a_i - 1$ 。
 - 若 $i - l$ 为偶数，则 a_i 的值增加 1，即 $a_i \leftarrow a_i + 1$ 。

现在，Yuki 想进行若干次操作，使得序列 a 中的所有数均相等。你需要帮助 Yuki 求出，使序列 a 中的所有数均相等的最小操作次数，或报告无解。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$)，表示测试数据组数。

对于每组测试数据：

- 第一行包含一个正整数 n ($1 \leq n \leq 10^6$)。
- 第二行包含 n 个整数 a_1, \dots, a_n ($0 \leq a_i \leq 10^{12}$)。

保证所有测试数据中 n 的总和不超过 10^6 。

Output

对于每组测试数据，输出一行：

- 若无解，则输出一个整数 -1 。
- 若有解，则输出一个整数，表示使序列 a 中的所有数均相等的最小操作次数。

Example

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

Note

对于第 1 组测试数据：

- 第 1 次操作选择区间 $[1, 2]$ 进行操作，原序列变为 $2, 2$ ，此时所有数都相同。



- 可以证明, 不存在操作次数更少的操作方案, 因此答案为 1。

对于第 2 组测试数据:

- 第 1 次操作选择区间 $[1, 4]$ 进行操作, 原序列变为 2, 4, 2, 4。
- 第 2 次操作选择区间 $[1, 4]$ 进行操作, 原序列变为 3, 3, 3, 3, 此时所有数都相同。
- 可以证明, 不存在操作次数更少的操作方案, 因此答案为 2。

对于第 3 组测试数据:

- 容易证明该序列在任意次操作内都无法使得所有数均相等, 故无解。



Problem B. Operating Robot

Input file: standard input
Output file: standard output

在一个平面直角坐标系上有一个机器人。初始时机器人位于 $(0, 0)$, Yuki 想要通过一系列指令使机器人到达 (x, y) 。

具体而言, 一个指令串为仅包含 01 的字符串:

- 0 表示向右移动一步, 即令机器人的位置由 (a, b) 变为 $(a + 1, b)$ 。
- 1 表示向上移动一步, 即令机器人的位置由 (a, b) 变为 $(a, b + 1)$ 。

现在, Yuki 有一个仅包含 012 的长度为 n 的指令串 $s = s_1 \dots s_n$ 。Yuki 需要先将这个指令串的 2 都替换成 0 或 1, 然后机器人会按照如下规则进行操作:

- 对于每个非负整数 i , 在第 i 秒时, 若机器人不位于 (x, y) , 则机器人会执行指令串的第 $((i \bmod n) + 1)$ 个指令。

Yuki 希望找到一种替换方式, 使得在机器人能够到达 (x, y) 的基础上, 指令串的字典序尽可能小。你需要帮助 Yuki 求出, 字典序最小的满足条件的替换后的指令串, 或报告不存在满足条件的指令串。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$), 表示测试数据组数。

对于每组测试数据:

- 第一行包含三个整数 n, x, y ($1 \leq n \leq 10^6, 0 \leq x, y \leq 10^{18}$)。
- 第二行包含一个长度为 n 的字符串 s ($s_i \in \{0, 1, 2\}$)。

保证所有测试数据中 n 的总和不超过 10^6 。

Output

对于每组测试数据, 输出一行:

- 若不存在满足条件的指令串, 则输出一个整数 -1 。
- 若存在满足条件的指令串, 则输出一个长度为 n 的字符串, 表示字典序最小的满足条件的替换后的指令串。



Example

standard input	standard output
6	01111
5 2 4	00111
01111	-1
5 3 3	011001
02221	-1
5 3 3	00100
00022	
6 1 0	
011201	
4 8 7	
2020	
5 0 0	
22102	

Note

对于第 1 组测试数据:

- 初始时机器人位于 $(0,0)$, 指令串为 01111。
- 按照操作规则, 机器人会依次移动到 $(1,0), (1,1), (1,2), (1,3), (1,4), (2,4)$ 。
- 由于机器人到达了 $(2,4)$, 故 01111 是一个满足条件的替换后的指令串。可以证明, 01111 是字典序最小的满足条件的替换后的指令串, 因此答案即为 01111。

对于第 2 组测试数据:

- 初始时机器人位于 $(0,0)$, 我们将指令串 02221 替换为 00111。
- 按照操作规则, 机器人会依次移动到 $(1,0), (2,0), (2,1), (2,2), (2,3), (3,3)$ 。
- 由于机器人到达了 $(3,3)$, 故 00111 是一个满足条件的替换后的指令串。可以证明, 00111 是字典序最小的满足条件的替换后的指令串, 因此答案即为 00111。

对于第 3 组测试数据:

- 可以证明, 不存在任意一种指令串的替换方式能够使得机器人到达 $(3,3)$ 。



Problem C. Palindromic and Balanced

Input file: standard input
Output file: standard output

Yuki 发现一个回文括号串不可能是平衡括号串，因此她设计了另一种定义回文平衡括号串的方式。

Yuki 按照如下规则定义了 **回文括号串**：

- 空串是回文括号串。
- (和) 是回文括号串。
- 若括号串 s 是回文括号串，则 $(s(和)s)$ 是回文括号串。

Yuki 按照如下规则定义了 **平衡括号串**：

- 空串是平衡括号串。
- 若括号串 s 是平衡括号串，则 (s) 是平衡括号串。
- 若括号串 s, t 均是平衡括号串，则 st (两括号串拼接起来) 是平衡括号串。

对于一个括号串 $s = s_1 \dots s_n$ ，Yuki 定义 s 是 **回文平衡括号串**，当且仅当：

- $s_2 \dots s_{n-1}$ 是回文括号串。
- $s_1 \dots s_n$ 是平衡括号串。

特殊地，空串和 () 也为回文平衡括号串。

例如，(())() 和 ()()()() 是回文平衡括号串，而 (((()))) 和 ()()()() 不是。

现在，Yuki 有一个长度为 n 的括号串 s ，她希望找到一个 s 的最长的子序列 $*$ ，使得该子序列是回文平衡括号串。不过 Yuki 并不知道怎么做，因此你需要帮助她求出 s 的最长的满足条件的子序列的长度。

*：称序列 a 是序列 b 的子序列，当且仅当序列 a 能够通过从序列 b 的基础上删除任意个元素（可以为 0 个）得到；特殊地，空序列是任何序列的子序列。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 5000$)，表示测试数据组数。

对于每组测试数据：

- 第一行包含一个正整数 n ($1 \leq n \leq 5000$)。
- 第二行包含一个长度为 n 的括号串 s ($s_i \in \{ (,) \}$)。

保证所有测试数据中 n 的总和不超过 10^4 。

Output

对于每组测试数据，输出一行，包含一个整数，表示 s 的最长的满足条件的子序列的长度。



Example

standard input	standard output
3	2
5	0
(((((6
7	
))))(((
8	
(((((

Note

对于第 1 组测试数据:

- 最长的满足条件的子序列为 $s_1s_3 = ()$ 和 $s_2s_3 = ()$, 答案为 2。

对于第 2 组测试数据:

- 最长的满足条件的子序列为空串, 答案为 0。

对于第 3 组测试数据:

- 最长的满足条件的子序列为 $s_1s_2s_4s_5s_6s_8 = ((()))$, 答案为 6。



Problem D. Qenerals

Input file: standard input
Output file: standard output

Yuki 在玩一款叫做 Qenerals 的游戏。

第 0 秒时, Yuki 拥有的士兵数量 $x = 0$, 占领的堡垒数量 $y = 1$ 。地图上还有 n 个未被占领的堡垒, 第 i 个堡垒的参数为 a_i 。

游戏一共会进行 m 秒。对于每个不大于 m 的正整数 i :

- 在第 i 秒开始时, 每个 Yuki 所占领的堡垒都会为 Yuki 生产 1 个士兵, 即 $x \leftarrow x + y$ 。
- 在第 i 秒结束时, Yuki 可以进行任意次操作 (可以为 0 次); 每次操作, Yuki 需要选择一个未被占领的堡垒 j 满足 $a_j \leq x$, 接着消耗 a_j 个士兵并占领第 j 个堡垒, 即 $x \leftarrow x - a_j$ 且 $y \leftarrow y + 1$ 。

你需要帮助 Yuki 求出, 在游戏结束后, 她最多能拥有多少个士兵。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$), 表示测试数据组数。

对于每组测试数据:

- 第一行包含两个正整数 n, m ($1 \leq n \leq 5 \cdot 10^5, 1 \leq m \leq 10^9$)。
- 第二行包含 n 个正整数 a_1, \dots, a_n ($1 \leq a_i \leq 10^9$)。

保证所有测试数据中 n 的总和不超过 $5 \cdot 10^5$ 。

Output

对于每组测试数据, 输出一行, 包含一个整数, 表示在游戏结束后 Yuki 最多能拥有的士兵数量。

Example

standard input	standard output
3	4
2 3	13
2 1	12
3 6	
1 1 3	
3 5	
1 1 1	

Note

对于第 1 组测试数据:

- 在第 1 秒开始时, Yuki 占领的堡垒数 $y = 1$, 因此 Yuki 拥有的士兵数量 x 由 0 变为了 1。
- 在第 1 秒结束时, Yuki 可以选择占领第 2 个堡垒, 因此 Yuki 占领的堡垒数 y 由 1 变为了 2, 拥有的士兵数量 x 由 1 变为了 0。



- 在第 2 秒开始时, Yuki 占领的堡垒数 $y = 2$, 因此 Yuki 拥有的士兵数量 x 由 0 变为了 2。
- 在第 2 秒结束时, Yuki 可以选择不进行任何操作。
- 在第 3 秒开始时, Yuki 占领的堡垒数 $y = 2$, 因此 Yuki 拥有的士兵数量 x 由 2 变为了 4。
- 在第 3 秒结束时, Yuki 可以选择不进行任何操作。
- 在游戏结束后, Yuki 拥有的士兵数量 x 达到了 4。可以证明, 在游戏结束后 Yuki 最多能拥有的士兵数量即为 4。

对于第 2 组测试数据:

- Yuki 可以在第 1, 2, 3 秒时分别占领第 1, 2, 3 个堡垒, 使 Yuki 在游戏结束后拥有的士兵数量达到 13。

对于第 3 组测试数据:

- Yuki 可以在第 1 秒时占领第 1 个堡垒, 并在第 2 秒时占领第 2 个堡垒和第 3 个堡垒, 使 Yuki 在游戏结束后拥有的士兵数量达到 12。



Problem E. Registration

Input file: standard input
Output file: standard output

ICPC 2026 长安邀请赛即将开赛。现有 n 支队伍打算参加比赛，其中第 i 支队伍会在第 v_i 秒时访问报名网页。

对于所有正整数 s ：

- 若在第 s 秒时访问报名网页的队伍数量不超过 x ，则所有在第 s 秒时访问报名网页的队伍报名成功。
- 否则，报名网页会因为太多队伍访问而在这一秒崩溃，所有在第 s 秒时访问报名网页的队伍报名失败。

作为 ICPC 2026 长安邀请赛的裁判长，Yuki 想要知道这场比赛的报名情况。因此，你需要帮助她计算报名成功的队伍数量。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$)，表示测试数据组数。

对于每组测试数据：

- 第一行包含两个正整数 n, x ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq x \leq n$)。
- 第二行包含 n 个正整数 v_1, \dots, v_n ($1 \leq v_i \leq 10^9$)。

保证所有测试数据中 n 的总和不超过 $2 \cdot 10^5$ 。

Output

对于每组测试数据，输出一行，包含一个整数，表示报名成功的队伍数量。

Example

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

Note

对于第 1 组测试数据：

- 第 1 秒时第 1, 4, 5 支队伍访问了报名网页， $3 > 2$ ，因此网页在这一秒崩溃了，这 3 支队伍报名失败。



- 第 2 秒时第 2 支队伍访问了报名网页, $1 < 2$, 因此网页没有崩溃, 这支队伍报名成功。
- 第 3 秒时第 3 支队伍访问了报名网页, $1 < 2$, 因此网页没有崩溃, 这支队伍同样报名成功。
- 一共有 2 支队伍报名成功, 因此答案为 2。

对于第 2 组测试数据:

- 网页在每一秒都没有崩溃, 5 支队伍全部报名成功, 因此答案为 5。

对于第 3 组测试数据:

- 网页在前 3 秒都崩溃了, 6 支队伍全部报名失败, 因此答案为 0。



Problem F. Split Sticks

Input file: standard input
Output file: standard output

Yuki 的面前有 n 根木棍排成一排, 第 i 根木棍的长度为 a_i 。

Yuki 定义一次操作为:

- 选择一根木棍, 并将其切成长度均为整数的两部分, 其中一部分的长度可以为 0。
- 将切成的左半部分木棍合并到这根木棍左边的第一个木棍; 若其左边没有木棍, 则左半部分木棍单独作为一根新的木棍。
- 将切成的右半部分木棍合并到这根木棍右边的第一个木棍; 若其右边没有木棍, 则右半部分木棍单独作为一根新的木棍。
- 删除所有长度为 0 的木棍。

现在, Yuki 想进行若干次操作, 使得所有木棍的长度均相等。你需要帮助 Yuki 求出, 使所有木棍的长度均相等所需的最小操作次数。

可以证明, 一定存在至少一种操作方案能够使所有木棍的长度均相等。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$), 表示测试数据组数。

对于每组测试数据:

- 第一行包含一个正整数 n ($1 \leq n \leq 10^6$)。
- 第二行包含 n 个正整数 a_1, \dots, a_n ($1 \leq a_i \leq 10^6$)。

保证所有测试数据中 n 的总和不超过 10^6 。

Output

对于每组测试数据, 输出一行, 包含一个整数, 表示使所有木棍的长度均相等所需的最小操作次数。

Example

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

Note

对于第 1 组测试数据:



- 第 1 次操作选择第 2 根木棍进行操作, 将其分成长度为 4, 1 的两段, 此时从左到右的木棍长度分别为 5, 5, 所有木棍的长度均相等。
- 可以证明使所有木棍的长度均相等所需的最小操作次数为 1 次。

对于第 2 组测试数据:

- 第 1 次操作选择第 1 根木棍进行操作, 将其分成长度为 0, 1 的两段, 此时从左到右的木棍长度分别为 5, 2, 5。
- 第 2 次操作选择第 2 根木棍进行操作, 将其分成长度为 1, 1 的两段, 此时从左到右的木棍长度分别为 6, 6, 所有木棍的长度均相等。
- 可以证明使所有木棍的长度均相等所需的最小操作次数为 2 次。

对于第 3 组测试数据:

- 初始时所有木棍的长度均相等, 故使所有木棍的长度均相等所需的最小操作次数为 0 次。



Problem G. Transform

Input file: standard input
Output file: standard output

Yuki 有一个大小为 n 的可重集合 $S = \{s_1, \dots, s_n\}$ 和一个整数 k 。

Yuki 定义一次变换为：

- 选择 S 的一个子集 S' (S' 可以为空集), 将 S' 从 S 中删除, 并将 S' 的 mex^* 添加到 S 中。

现在, Yuki 想进行若干次变换, 使得 S 变为 $\{k\}$ 。你需要帮助 Yuki 求出, 使 S 变为 $\{k\}$ 所需的最小变换次数。由于答案可能很大, 你只需要输出答案对 998244353 取模的结果即可。

可以证明, 一定存在至少一种操作方案能够使 S 变为 $\{k\}$ 。

*: 一个可重集的 mex 为该可重集中未出现过的最小非负整数, 例如 $\text{mex}\{0, 1, 2\} = 3$, $\text{mex}\{1, 0, 3, 1\} = 2$, $\text{mex}\emptyset = 0$ 。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$), 表示测试数据组数。

对于每组测试数据：

- 第一行包含两个整数 n, k ($1 \leq n \leq 5 \cdot 10^5$, $0 \leq k \leq 10^9$)。
- 第二行包含 n 个整数 s_1, \dots, s_n ($0 \leq s_i \leq 10^9$)。

保证所有测试数据中 n 的总和不超过 $5 \cdot 10^5$ 。

Output

对于每组测试数据, 输出一行, 包含一个整数, 表示使 S 变为 $\{k\}$ 所需的最小变换次数对 998244353 取模的结果。

Example

standard input	standard output
6	2
1 2	0
1	3
1 4	2
4	1
3 3	262875292
0 2 2	
4 2	
1 0 3 2	
4 3	
2 1 0 2	
3 52	
20 2 6	



Note

对于第 1 组测试数据:

- Yuki 可以在第 1 次变换中选择 $S' = \emptyset$, 使 S 变为 $\{0, 1\}$, 再在第 2 次变换中选择 $S' = \{0, 1\}$, 使 S 变为 $\{2\}$ 。
- 可以证明, 不存在变换次数更少的操作方案, 因此答案为 2。

对于第 2 组测试数据:

- Yuki 不需要进行变换即可使 $S = \{4\}$, 因此答案为 0。

对于第 3 组测试数据:

- Yuki 可以在第 1 次变换中选择 $S' = \emptyset$, 使 S 变为 $\{0, 0, 2, 2\}$, 在第 2 次变换中选择 $S' = \{0, 2\}$, 使 S 变为 $\{0, 1, 2\}$, 再在第 3 次变换中选择 $S' = \{0, 1, 2\}$, 使 S 变为 $\{3\}$ 。
- 可以证明, 不存在变换次数更少的操作方案, 因此答案为 3。

对于第 4 组测试数据:

- Yuki 可以在第 1 次变换中选择 $S' = \{2, 3\}$, 使 S 变为 $\{0, 0, 1\}$, 再在第 2 次变换中选择 $S' = \{0, 0, 1\}$, 使 S 变为 $\{2\}$ 。
- 可以证明, 不存在变换次数更少的操作方案, 因此答案为 2。

对于第 5 组测试数据:

- Yuki 可以直接在第 1 次变换中选择 $S' = \{0, 1, 2, 2\}$, 使 S 变为 $\{3\}$ 。
- 可以证明, 不存在变换次数更少的操作方案, 因此答案为 1。



Problem H. Unreachable Land

Input file: standard input
Output file: standard output

Yuki 梦想前往那片不可到达之地, 经过多年的努力, 她的面前只剩下了这样一道题目。

给定三个整数 a, b, m 。你需要进行 m 轮操作, 第 i 轮操作可以令 $a \leftarrow a \bmod (m - i + 1)$ 或者不进行修改。求 m 轮操作后 $a = b$ 的方案数, 答案对 998244353 取模。

定义两种方案不同, 当且仅当存在 $1 \leq i \leq m$, 使得一种方案中第 i 轮进行了修改, 而另一种方案中第 i 轮没有进行修改。注意, 只要选择执行 $a \leftarrow a \bmod (m - i + 1)$ 即视为进行了修改, 不论修改后 a 的值是否变化。

你曾经也幻想登上只存在于童话里的不可到达之地, 如今 Yuki 有机会实现这个梦想, 你必须帮助她。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$), 表示测试数据组数。

对于每组测试数据:

- 共一行, 包含三个整数 a, b, m ($0 \leq b < m \leq a \leq 2 \cdot 10^5$)。

保证所有测试数据的 a 之和不超过 $2 \cdot 10^5$ 。

Output

对于每组测试数据, 输出一行, 包含一个整数, 表示答案对 998244353 取模的结果。

Example

standard input	standard output
5	25
5 0 5	1
5 2 3	14
10 1 7	0
10 6 10	837481226
100000 114 514	

Note

对于第 1 组测试数据:

- 其中一种满足要求的操作方案为, 在第 3 轮操作中和第 4 轮操作中进行修改。
- 另一种满足要求的操作方案为, 在第 1, 2, 3, 4, 5 轮操作中均进行修改。

对于第 2 组测试数据:

- 唯一一种满足要求的操作方案为, 在第 3 轮操作中进行修改。

对于第 4 组测试数据:

- 可以证明不存在满足要求的操作方案。



Problem I. VIP Coupon

Input file: standard input
Output file: standard output

长安城给 Yuki 留下了许多美好的印象, 因此 Yuki 来到了一家商店, 打算买一些纪念品带回家。

这家商店一共出售 n 个纪念品和 m 张 VIP 优惠券, 第 i 个纪念品的价格为 a_i , 第 j 张 VIP 优惠券的价格为 b_j , 参数为 c_j 。其中, 一张参数为 v 的 VIP 优惠券的效果为:

- 设购买该优惠券后, 下一次购买的物品 (包括纪念品和 VIP 优惠券) 的价格为 x , 则该物品的价格会变为 $\max(x - v, 0)$ 。

VIP 优惠券的效果会强制在下次购买时生效, 不能自主选择使用时间。显然, 根据此规则, VIP 优惠券的效果也无法叠加。每个物品 (包括纪念品和 VIP 优惠券) 只能购买至多一次, 不能重复购买。

现在, Yuki 打算按照任意顺序购买所有纪念品和任意张 VIP 优惠券 (可以为 0 张)。你需要帮助 Yuki 求出, 买下所有纪念品的最小花费。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$), 表示测试数据组数。

对于每组测试数据:

- 第一行包含两个正整数 n, m ($1 \leq n, m \leq 5 \cdot 10^5$)。
- 第二行包含 n 个整数 a_1, \dots, a_n ($0 \leq a_i \leq 10^9$)。
- 第三行包含 m 个整数 b_1, \dots, b_m ($0 \leq b_i \leq 10^9$)。
- 第四行包含 m 个整数 c_1, \dots, c_m ($0 \leq c_i \leq 10^9$)。

保证所有测试数据中 n 和 m 的总和均不超过 $5 \cdot 10^5$ 。

Output

对于每组测试数据, 输出一行, 包含一个整数, 表示买下所有纪念品的最小花费。

Example

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

Note

对于第 1 组测试数据:



- Yuki 可以依次购买第 1 张优惠券, 第 1 个纪念品, 第 3 张优惠券, 第 2 个纪念品。
- 在优惠后, 第 1 个纪念品的价格变为了 0, 第 2 个纪念品的价格变为了 1, 总花费为 $1+0+2+1=4$ 。

对于第 2 组测试数据:

- Yuki 可以依次购买第 1 个纪念品, 第 1 张优惠券, 第 2 个纪念品, 第 3 张优惠券, 第 2 张优惠券, 第 3 个纪念品。
- 在优惠后, 第 2 个纪念品的价格变为了 0, 第 2 张优惠券的价格变为了 0, 第 3 个纪念品的价格变为了 1, 总花费为 $2+0+0+2+0+1=5$ 。



Problem J. Would You Make a Convex?

Input file: standard input
Output file: standard output

Yuki 是国际凸多边形锦标赛 (International Convex Polygon Championship, ICPC) 的裁判长。他为比赛提出了一道几何题。然而, 由于他在几何方面经验不足, 未能生成正确的凸多边形数据。

为了证明自己的几何能力, Yuki 再次开始玩起了木棍。他有 n 根木棍, 第 i 根木棍的长度为 a_i 。他打算从中选出至少 3 根木棍, 使得这些木棍可以组成一个非退化凸多边形*。

然而, 由于 Yuki 并不了解几何方面的知识, 他不知道应该如何选择木棍。作为 Yuki 的好朋友, 你需要帮助他找到木棍的一个子集, 使得:

- 该子集中包含至少 3 根木棍, 且包含的木棍数量尽可能多;
- Yuki 从该子集中 **任意** 选出至少 3 根木棍, 这些木棍都可以组成一个非退化凸多边形。

或报告不存在满足要求的子集。

*: 所有边长大于零、没有三点共线且所有内角严格小于 180° 的凸多边形被称作非退化凸多边形。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$), 表示测试数据组数。

对于每组测试数据:

- 第一行包含一个正整数 n ($3 \leq n \leq 5 \cdot 10^5$)。
- 第二行包含 n 个正整数 a_1, \dots, a_n ($1 \leq a_i \leq 10^9$)。

保证所有测试数据中 n 的总和不超过 $5 \cdot 10^5$ 。

Output

对于每组测试数据, 输出一行:

- 若不存在满足要求的子集, 则输出一个整数 0。
- 若存在满足要求的子集, 则先输出一个整数 k , 表示你找到的子集的大小, 再输出 k 个整数 b_1, \dots, b_k , 表示你找到的子集中的 k 根木棍的长度。

Example

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



Note

对于第 1 组测试数据:

- $\{6, 2, 6\}$ 为满足要求的子集; 由于 $2 + 6 > 6$, 这 3 根木棍可以组成一个非退化三角形; 容易证明不存在更大的满足要求的子集。
- $\{6, 6, 9\}$ 同样为满足要求的子集。

对于第 2 组测试数据:

- 可以证明不存在满足要求的子集。

对于第 3 组测试数据:

- $\{3, 4, 5, 6\}$ 为满足要求的子集; 此时 Yuki 有 $\{3, 4, 5\}, \{3, 4, 6\}, \{3, 5, 6\}, \{4, 5, 6\}, \{3, 4, 5, 6\}$ 共 5 种选择木棍的方式, 而每种选择方式都可以组成一个非退化凸多边形; 容易证明不存在更大的满足要求的子集。
- $\{3, 4, 5, 6, 9\}$ 不为满足要求的子集, 因为长度为 3, 4, 9 的木棍无法组成一个非退化三角形。
- $\{3, 5, 6\}$ 不为满足要求的子集, 因为存在更大的满足要求的子集。



Problem K. XOR and LCA

Input file: standard input
Output file: standard output

Yuki 有一棵包含 2^n 个结点的树，结点的编号为 0 至 $2^n - 1$ ，第 i 条边连接结点 u_i 与结点 v_i 。
设 $\text{lca}_r(u, v)$ 表示，以结点 r 为根时，结点 u 与结点 v 的最近公共祖先。你需要帮助 Yuki 求出：

$$\bigoplus_{0 \leq u < v < 2^n} \text{lca}_{u \oplus v}(u, v)$$

其中 \oplus 表示按位异或运算。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^4$)，表示测试数据组数。

对于每组测试数据：

- 第一行包含一个正整数 n ($1 \leq n \leq 21$)。
- 接下来 $2^n - 1$ 行，第 i 行包含两个正整数 u_i, v_i ($0 \leq u_i, v_i < 2^n$, $u_i \neq v_i$)。

保证输入数据形成一棵树，保证所有测试数据中 2^n 的总和不超过 2^{21} 。

Output

对于每组测试数据，输出一行，包含一个整数，表示答案。



Example

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

Note

对于第 1 组测试数据:

- 以结点 1 为根时, 结点 0 与结点 1 的最近公共祖先为结点 1, 因此答案为 $\text{lca}_1(0, 1) = 1$ 。

对于第 2 组测试数据:

- 我们计算所有点对 (u, v) 的 $\text{lca}_{u \oplus v}(u, v)$:
 - $(0, 1)$: $0 \oplus 1 = 1$, $\text{lca}_1(0, 1) = 1$ 。
 - $(0, 2)$: $0 \oplus 2 = 2$, $\text{lca}_2(0, 2) = 2$ 。
 - $(0, 3)$: $0 \oplus 3 = 3$, $\text{lca}_3(0, 3) = 3$ 。
 - $(1, 2)$: $1 \oplus 2 = 3$, $\text{lca}_3(1, 2) = 2$ 。
 - $(1, 3)$: $1 \oplus 3 = 2$, $\text{lca}_2(1, 3) = 2$ 。
 - $(2, 3)$: $2 \oplus 3 = 1$, $\text{lca}_1(2, 3) = 2$ 。
- 异或和为 $1 \oplus 2 \oplus 3 \oplus 2 \oplus 2 \oplus 2 = 2$ 。



Problem L. Yesterday Once More (Easy Version)

Input file: standard input
Output file: standard output

这是本题的简单版本。简单版本和困难版本的唯一区别在于你给出的方案的移动次数限制。

Yuki 生活在一个 $n + 1$ 行 n 列的棋盘上。棋盘从上至下依次为第 1 行至第 $n + 1$ 行，从左至右依次为第 1 列至第 n 列。设 (i, j) 表示棋盘上第 i 行第 j 列的格子。

棋盘上共有 $n - 1$ 个格子中有障碍，且这些障碍的分布满足：

- 第 1 行和第 $n + 1$ 行中没有障碍。
- 对于所有 $2 \leq i \leq n$ ，第 i 行中有 **恰好** 一个障碍。
- 对于所有 $1 \leq j \leq n$ ，第 j 列中有 **至多** 一个障碍。

初始时，Yuki 位于 $(1, 1)$ ；她听说棋盘的第 $n + 1$ 行生活着一群袋鼠，因此她想去棋盘的第 $n + 1$ 行，看看那边的风景。

为了实现目标，Yuki 可以进行若干次移动。每次移动，她需要选定上下左右中的一个方向，并向该方向移动一个格子。特殊地，若该格子位于棋盘外或该格子中有障碍，则此次移动不会被执行。

糟糕的是，Yuki 只知道障碍的分布规则，并不知道障碍的具体分布方式。因此，她希望你帮助她指定每次移动的方向，使得对于任意满足要求的障碍分布方式，Yuki 都 **到达过** 棋盘的第 $n + 1$ 行（她只希望她到达过第 $n + 1$ 行就好，不需要保证在所有移动结束后 Yuki 仍位于第 $n + 1$ 行）。

由于 Yuki 没那么着急，你给出的方案的移动次数不能大于 $30 \cdot n$ 。

Input

共一行，包含一个正整数 n ($2 \leq n \leq 10^3$)。

Output

第一行，输出一个整数 k ($1 \leq k \leq 30 \cdot n$)，表示你给出的方案的移动次数。

第二行，输出一个长度为 k 的字符串 s ，其中 s_i 表示第 i 次移动中 Yuki 的移动方向：

- 若 $s_i = U$ ，则表示第 i 次移动中 Yuki 的移动方向为向上。
- 若 $s_i = D$ ，则表示第 i 次移动中 Yuki 的移动方向为向下。
- 若 $s_i = L$ ，则表示第 i 次移动中 Yuki 的移动方向为向左。
- 若 $s_i = R$ ，则表示第 i 次移动中 Yuki 的移动方向为向右。

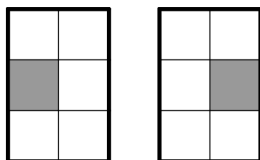
Examples

standard input	standard output
2	4 DRDD
3	17 DDDUUURDDDUUURDDD

Note

对于第 1 组样例:

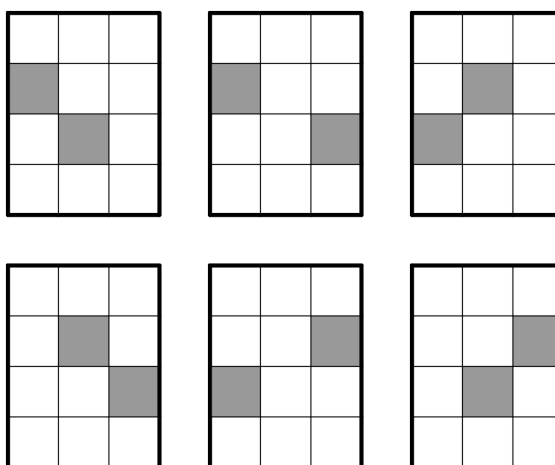
- 设灰色格子表示有障碍的格子, 白色格子表示没有障碍的格子, 则下图给出了所有满足要求的障碍分布方式:



- 对于第 1 种障碍分布方式, Yuki 的移动路径为 $(1,1) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (3,2)$ 。
- 对于第 2 种障碍分布方式, Yuki 的移动路径为 $(1,1) \rightarrow (2,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,1)$ 。
- 对于每种满足要求的障碍分布方式, Yuki 都到达过棋盘的第 $n+1$ 行, 因此样例输出正确。

对于第 2 组样例:

- 设灰色格子表示有障碍的格子, 白色格子表示没有障碍的格子, 则下图给出了所有满足要求的障碍分布方式:



- 容易证明, 对于其中任意一种障碍分布方式, 按照样例输出中给出的移动方式移动, Yuki 都到达过棋盘的第 $n+1$ 行。



Problem M. Yesterday Once More (Hard Version)

Input file: standard input
Output file: standard output

这是本题的困难版本。简单版本和困难版本的唯一区别在于你给出的方案的移动次数限制。

Yuki 生活在一个 $n + 1$ 行 n 列的棋盘上。棋盘从上至下依次为第 1 行至第 $n + 1$ 行，从左至右依次为第 1 列至第 n 列。设 (i, j) 表示棋盘上第 i 行第 j 列的格子。

棋盘上共有 $n - 1$ 个格子中有障碍，且这些障碍的分布满足：

- 第 1 行和第 $n + 1$ 行中没有障碍。
- 对于所有 $2 \leq i \leq n$ ，第 i 行中有 **恰好** 一个障碍。
- 对于所有 $1 \leq j \leq n$ ，第 j 列中有 **至多** 一个障碍。

初始时，Yuki 位于 $(1, 1)$ ；她听说棋盘的第 $n + 1$ 行生活着一群袋鼠，因此她想去棋盘的第 $n + 1$ 行，看看那边的风景。

为了实现目标，Yuki 可以进行若干次移动。每次移动，她需要选定上下左右中的一个方向，并向该方向移动一个格子。特殊地，若该格子位于棋盘外或该格子中有障碍，则此次移动不会被执行。

糟糕的是，Yuki 只知道障碍的分布规则，并不知道障碍的具体分布方式。因此，她希望你帮助她指定每次移动的方向，使得对于任意满足要求的障碍分布方式，Yuki 都 **到达过** 棋盘的第 $n + 1$ 行（她只希望她到达过第 $n + 1$ 行就好，不需要保证在所有移动结束后 Yuki 仍位于第 $n + 1$ 行）。

由于 Yuki 十分着急，你给出的方案的移动次数不能大于 $10 \cdot n$ 。

Input

共一行，包含一个正整数 n ($2 \leq n \leq 10^3$)。

Output

第一行，输出一个整数 k ($1 \leq k \leq 10 \cdot n$)，表示你给出的方案的移动次数。

第二行，输出一个长度为 k 的字符串 s ，其中 s_i 表示第 i 次移动中 Yuki 的移动方向：

- 若 $s_i = U$ ，则表示第 i 次移动中 Yuki 的移动方向为向上。
- 若 $s_i = D$ ，则表示第 i 次移动中 Yuki 的移动方向为向下。
- 若 $s_i = L$ ，则表示第 i 次移动中 Yuki 的移动方向为向左。
- 若 $s_i = R$ ，则表示第 i 次移动中 Yuki 的移动方向为向右。

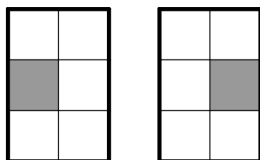
Examples

standard input	standard output
2	4 DRDD
3	17 DDDUUURDDDUUURDDD

Note

对于第 1 组样例:

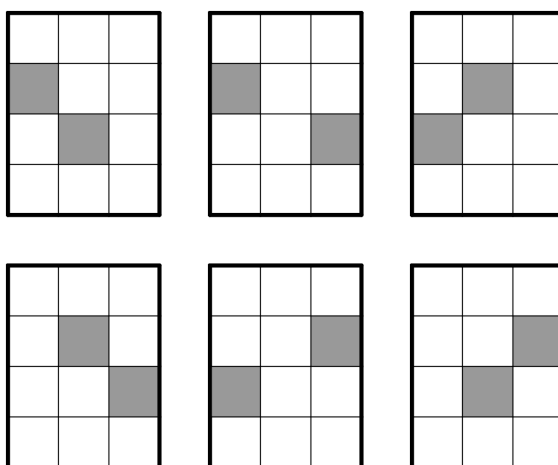
- 设灰色格子表示有障碍的格子, 白色格子表示没有障碍的格子, 则下图给出了所有满足要求的障碍分布方式:



- 对于第 1 种障碍分布方式, Yuki 的移动路径为 $(1,1) \rightarrow (1,1) \rightarrow (1,2) \rightarrow (2,2) \rightarrow (3,2)$ 。
- 对于第 2 种障碍分布方式, Yuki 的移动路径为 $(1,1) \rightarrow (2,1) \rightarrow (2,1) \rightarrow (3,1) \rightarrow (3,1)$ 。
- 对于每种满足要求的障碍分布方式, Yuki 都到达过棋盘的第 $n+1$ 行, 因此样例输出正确。

对于第 2 组样例:

- 设灰色格子表示有障碍的格子, 白色格子表示没有障碍的格子, 则下图给出了所有满足要求的障碍分布方式:



- 容易证明, 对于其中任意一种障碍分布方式, 按照样例输出中给出的移动方式移动, Yuki 都到达过棋盘的第 $n+1$ 行。



Problem N. Zebra Crossing

Input file: standard input
Output file: standard output

Yuki 的面前有一条奇怪的斑马线。

这条斑马线可以看作一棵包含 n 个结点的树，第 i 条边连接结点 u_i 与结点 v_i ，每个结点的颜色为黑色或白色，用一个长度为 n 的 01 串 s 描述：

- 若 $s_i = 0$ ，则结点 i 的颜色为黑色。
- 若 $s_i = 1$ ，则结点 i 的颜色为白色。

Yuki 有一个跳跃能力 k ，表示当她位于结点 x 时，她可以通过一次跳跃，移动到任意一个满足 $\text{dist}(x, y) \leq k$ 的结点 y 上。其中， $\text{dist}(x, y)$ 表示结点 x 到结点 y 的简单路径上的边的数量。

接下来，Yuki 会在斑马线上进行 $n - 1$ 轮跳跃。在第 i 轮跳跃中，Yuki 初始时位于结点 1，她希望通过若干次跳跃恰好移动到结点 $i + 1$ 。同时，Yuki 希望最小化她跳跃后踩到黑色结点的次数。

你需要帮助 Yuki 求出，每一轮跳跃中 Yuki 跳跃后踩到黑色结点的次数的最小值。

Input

本题包含多组测试数据。

第一行包含一个正整数 t ($1 \leq t \leq 10^5$)，表示测试数据组数。

对于每组测试数据：

- 第一行包含两个正整数 n, k ($1 \leq n \leq 5 \cdot 10^5$, $1 \leq k \leq n$)。
- 第二行包含一个长度为 n 的 01 串 s ($s_i \in \{0, 1\}$)。
- 接下来 $n - 1$ 行，第 i 行包含两个正整数 u_i, v_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$)。

保证所有测试数据中 n 的总和不超过 $5 \cdot 10^5$ 。

Output

对于每组测试数据，输出一行，包含 $n - 1$ 个整数，第 i 个整数表示第 i 轮跳跃中 Yuki 跳跃后踩到黑色结点的次数的最小值。



Example

standard input	standard output
2	0 1 1 2
5 1	1 1 1 0 1 1 1 2
01010	
3 5	
2 1	
1 3	
3 4	
9 3	
100010000	
1 2	
2 3	
2 4	
3 5	
3 6	
4 7	
6 8	
7 9	

Note

对于第 1 组测试数据:

- 对于第 1 轮跳跃, 一种满足要求的方案中经过的点依次为 1, 2。
- 对于第 4 轮跳跃, 一种满足要求的方案中经过的点依次为 1, 3, 5。

对于第 2 组测试数据:

- 对于第 4 轮跳跃, 一种满足要求的方案中经过的点依次为 1, 5。
- 对于第 7 轮跳跃, 一种满足要求的方案中经过的点依次为 1, 5, 8。
- 对于第 8 轮跳跃, 一种满足要求的方案中经过的点依次为 1, 4, 9。