

Problem A. 字符串序列

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

OnO 最近对最长上升子序列 (Longest Increasing Subsequence — LIS) 很感兴趣, 所以他尝试对字符串做 LIS:

他会把这个串划分为若干段形成一个字符串序列, 然后求出 LIS 的长度。

例如串 ababab 可以划分成字符串序列 a|bab|ab,

此时这个序列的 LIS 的长度是 2 (a|bab|ab)。

但是如果将串 ababab 被划分 a|ba|bab, 那么不难发现此时 LIS 的长度是 3。

OnO 现在有一个长度为 n 的字符串 s 。

OnO 希望他划分 s 得到的字符串序列的 LIS 长度尽可能大。

他想知道在所有可能的划分方式中, LIS 长度的最大值是多少?

Input

第一行输入一个数 n 表示字符串的长度。

输入一行, 一个长度为 n 的字符串, 表示 s 。

保证 s 只包含小写英文字母。

Output

输出一行, 一个数, 表示字符串 s 的所有可能的划分方式中, LIS 长度的最大值。

Scoring

本题采用捆绑测试。

- Subtask 1 (5 pts) : $n \leq 15$ 。
- Subtask 2 (10 pts) : $n \leq 100$ 。
- Subtask 3 (15 pts) : $n \leq 300$ 。
- Subtask 4 (30 pts) : $n \leq 1000$ 。
- Subtask 5 (40 pts) : 无特殊限制。

对于所有数据, $1 \leq n \leq 2 \times 10^4$, 保证字符串只由小写字母组成。

Examples

standard input	standard output
6 ababab	3
6 aaabbb	4
14 sixthturingcup	4

Note

对于第一个样例，一种最优的划分方法是 $a|b|ab|a|b$ 。

对于第二个样例，一种最优的划分方法是 $a|aa|b|bb$ 。

Problem B. 伪物

Input file: standard input
Output file: standard output
Time limit: 3.5 seconds
Memory limit: 128 megabytes

这个世上没有比假花更美丽的花。

《フォニイ》

在花束枯萎之后，昔日花下的回忆会在泥土中慢慢消散，最终成为记忆中不愿提起的部分。
有办法让那朵花变成假花吗？即使扭曲的情感在碰撞之后更加失真，也可以留下一些吧。
可是我已经没有时间去验证每一种可能了啊...也许只是尝试最想改变的几个瞬间，就不会有遗憾了。
给定长度为 n 的序列 a ，你需要回答 m 次询问。

每次询问将会给定 l, r ，你需要计算 $\bigoplus_{i=l}^r \bigoplus_{j=i}^r (a_i + a_j)$ 的值。

Input

第一行输入两个整数 n, m 。

第二行输入 n 个整数 a_i 。

接下来 m 行每行输入两个整数 l, r 。

Output

输出 m 行，每行包含一个整数，代表每个询问的答案。

Scoring

本题采用捆绑测试。

- Subtask 1 (10 pts) : $n, m \leq 500$ 。
- Subtask 2 (10 pts) : $n, m \leq 5 \times 10^3$ 。
- Subtask 3 (20 pts) : $n \leq 5 \times 10^4, m \leq 10^5$ 。
- Subtask 4 (15 pts) : $n \leq 10^5, m \leq 5 \times 10^5$ 。
- Subtask 5 (15 pts) : $m \leq 5 \times 10^5$ 。
- Subtask 6 (30 pts) : 无特殊限制。

对于所有数据， $1 \leq n \leq 2 \times 10^5$ ， $1 \leq m \leq 10^6$ ， $0 \leq a_i < 2^{18}$ ， $1 \leq l \leq r \leq n$ 。

请注意本题非常规的空间限制，和算法时间复杂度系数对实际运行时间的影响。

Example

standard input	standard output
4 4	10
3 1 4 2	14
1 3	0
2 4	10
1 2	
3 4	

Note

题目中的 \oplus 为异或求和符号。 \oplus 和求和符号 \sum 的唯一区别是 \oplus 计算其所有项的异或和而非数值和。
对于第一组询问，答案为 $(3 + 3) \oplus (3 + 1) \oplus (3 + 4) \oplus (1 + 1) \oplus (1 + 4) \oplus (4 + 4) = 10$ 。

Problem C. 最优移动

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

在一个平面直角坐标系的 x 轴上方，有一些人排成 n 个与 y 轴平行的队伍，一个三元组 (x_i, s_i, l_i) 表示 $(x_i, s_i), (x_i, s_i + 1) \dots (x_i, s_i + l_i - 1)$ 这 l_i 个位置上每个位置有一个人，这些人组成了第 i 个队伍（保证不存在两个队伍包含同一位置）。

你可以执行以下操作任意次：

- 选择一个人，将其平行坐标轴移动 1 个单位长度。

你需要保证任意时刻不能使有两个人在同一位置。

请求出将每个人都移动到 x 轴及以下需要的最小操作次数。

Input

第一行一个整数 n ，表示队伍的数量。

接下来 n 行，每行三个整数 x_i, s_i, l_i ， (x_i, s_i, l_i) 即为第 i 个队伍的三元组。

Output

一行一个整数，表示最小操作次数。

Scoring

本题采用捆绑测试。

- Subtask 1 (10 pts) : $\forall 1 \leq i < j \leq n, |x_i - x_j| \geq 500$ 。
- Subtask 2 (15 pts) : $|x_i| \leq 5 \times 10^2, \sum l_i \leq 2 \times 10^3$ 。
- Subtask 3 (15 pts) : $|x_i| \leq 2.5 \times 10^3, \sum l_i \leq 10^4$ 。
- Subtask 4 (15 pts) : $|x_i| \leq 10^5, \sum l_i \leq 4 \times 10^5, l_i \leq 10^4$ 。
- Subtask 5 (20 pts) : $n \leq 2 \times 10^4, l_i \leq 10^4$ 。
- Subtask 6 (25 pts) : 无特殊限制。

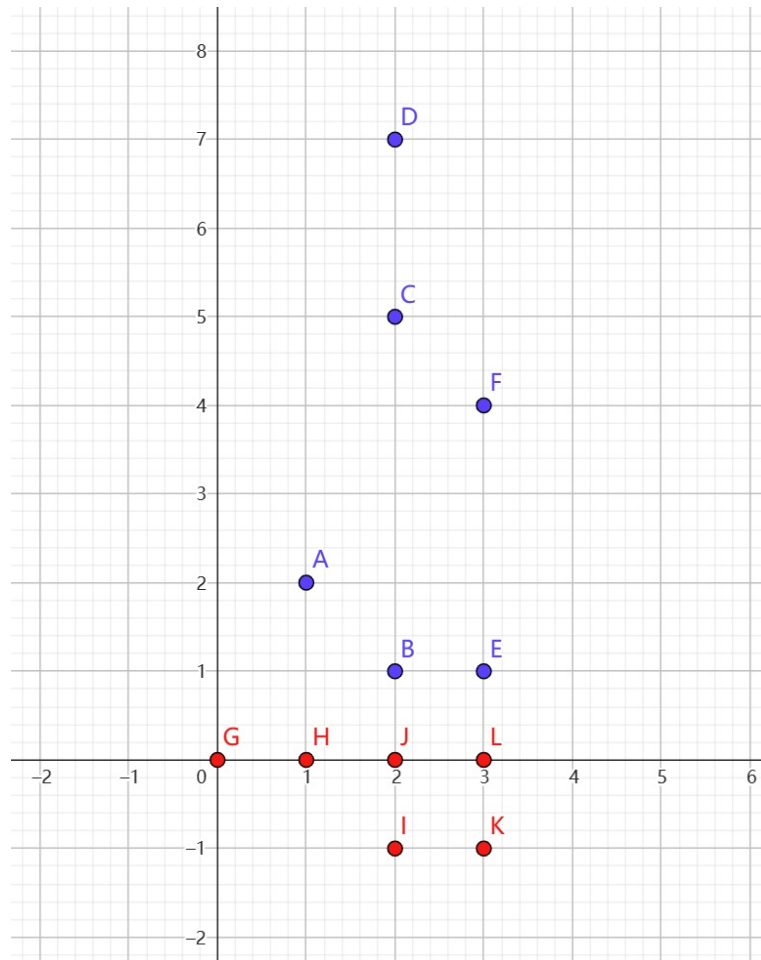
对于所有数据， $1 \leq n \leq 10^5$ ， $-10^9 \leq x_i \leq 10^9$ ， $1 \leq s_i + l_i - 1 \leq 10^9$ ， $1 \leq l_i \leq 5 \times 10^4$ 。

Example

standard input	standard output
6 1 2 1 2 1 1 2 5 1 2 7 1 3 1 1 3 4 1	24

Note

如下图，将 A 移动到 G ， B 移动到 H ， C 移动到 I ， D 移动到 J ， E 移动到 K ， F 移动到 L ，需要 24 次操作，并且可以证明不存在更优的操作方法。



Problem D. 合并游戏

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

Pub 正在玩合并游戏。

Pub 有一个大小为 n 的栈，初始为空。

当栈未满时，Pub 会向栈顶不断加入数。每次加入的数在 $\{1, 2\}$ 随机：有 $\frac{x}{y}$ 的概率加入一个 1， $1 - \frac{x}{y}$ 的概率加入一个 2。

在任意时刻，如果栈的大小 ≥ 2 ，且栈顶两个数相同，会发生一次合并：这两个数会立刻被弹出，并向栈中加入他们的和。

当栈满，且无法发生任何合并时，游戏结束。

下面给出 $n = 3$ 时可能的一个过程：

初始 $s = []$ 。

1. 加入 1，此时 $s = [1]$;
2. 加入 1，发生一次合并，此时 $s = [2]$;
3. 加入 1，此时 $s = [2, 1]$;
4. 加入 1，发生两次合并，此时 $s = [4]$;
5. 加入 2，此时 $s = [4, 2]$;
6. 加入 2，发生两次合并，此时 $s = [8]$;
7. 加入 1，此时 $s = [8, 1]$;
8. 加入 2，此时 $s = [8, 1, 2]$ ，栈满，游戏结束。

最终 $s = [8, 1, 2]$ 。可以证明，这个游戏一定会在有限步操作内停止。

Pub 想要知道游戏结束时，栈中所有数之和的期望。

Pub 会给你两个数 $op(0 \leq op \leq 1), m$ ，当 $op = 0$ 时，你需要求出 $n = m$ 时的答案；当 $op = 1$ 时，你需要对 $n = 1, 2, 3, \dots, m$ 都求出答案。

答案对 998244353 取模。

Input

一行四个整数 op, m, x, y 。

Output

若 $op = 0$ ，输出一行一个数代表 $n = m$ 的答案。

若 $op = 1$ ，输出一行 m 个数，第 i 个数代表 $n = i$ 的答案。

Scoring

本题采用捆绑测试。

- Subtask 1 (5 pts) : $n \leq 7$ 。
- Subtask 2 (10 pts) : $n \leq 20$ 。

- Subtask 3 (10 pts) : $op = 0, n \leq 500$ 。
- Subtask 4 (5 pts) : $n \leq 500$ 。
- Subtask 5 (10 pts) : $op = 0, n \leq 5000$ 。
- Subtask 6 (15 pts) : $n \leq 5000$ 。
- Subtask 7 (10 pts) : $n \leq 50000$ 。
- Subtask 8 (15 pts) : $op = 0, n \leq 100000$ 。
- Subtask 9 (20 pts) : 无特殊限制。

对于所有数据, $0 \leq op \leq 1, 1 \leq n \leq 100000, 1 \leq x < y \leq 998244352$ 。

Examples

standard input	standard output
0 1 1 2	499122178
1 2 1 2	499122178 62390276
1 4 1550 45997	937238966 555960604 114514 975068979

Note

$n = 1$ 时, 最终的栈有 $\frac{1}{2}$ 的概率是 [1], $\frac{1}{2}$ 的概率是 [2], 期望为 $\frac{3}{2}$ 。

$n = 2$ 时: 有以下几种情况:

- 加入顺序为 111, 栈为 [2, 1], 概率 $\frac{1}{8}$, 和为 3。
- 加入顺序为 1121, 栈为 [4, 1], 概率 $\frac{1}{16}$, 和为 5。
- 加入顺序为 1122, 栈为 [4, 2], 概率 $\frac{1}{16}$, 和为 6。
- 加入顺序为 12, 栈为 [1, 2], 概率 $\frac{1}{4}$, 和为 3。
- 加入顺序为 21, 栈为 [2, 1], 概率 $\frac{1}{4}$, 和为 3。
- 加入顺序为 221, 栈为 [4, 1], 概率 $\frac{1}{8}$, 和为 5。
- 加入顺序为 222, 栈为 [4, 2], 概率 $\frac{1}{8}$, 和为 6。

期望为 $3 \times \frac{1}{8} + 5 \times \frac{1}{16} + 6 \times \frac{1}{16} + 3 \times \frac{1}{4} + 3 \times \frac{1}{4} + 5 \times \frac{1}{8} + 6 \times \frac{1}{8} = \frac{63}{16}$ 。