

Problem A. 生日

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

昨天是 Iris 的生日，让我们一起祝她生日快乐！

oql 认为，如何获得他人的生日信息是个十分有趣的问题。但是，当你有了别人的身份证号后，知道他/她的生日信息不再是难题了！

众所周知，我国的身份证号码共 18 位，其中：

- 第 1 ~ 6 位表示所在地区。其中：第 1,2 位表示所在省份（自治区、直辖市、特别行政区）的代码；第 3,4 位表示市（地级市、自治州、地区、盟及直辖市所属区 and 县的汇总码）；第 5,6 位表示县（区、县级市、旗）。
- 第 7 ~ 14 位表示出生的年、月、日。其中：第 7 ~ 10 位表示年份，第 11,12 位表示出生月份，第 13,14 位表示出生的日期。
- 第 15 ~ 17 位为顺序码。其中第 17 位为奇数则为男性、为偶数则为女性。
- 第 18 位为校验码，为 0 ~ 9 或 X。

现在，oql 获得了 T 个身份证号，他需要依次求出这些身份证号所对应的生日。

生日的格式形如月份，日期，年份。其中，两个英文半角逗号后紧跟一个空格，月份使用英文简写进行表示，1 ~ 12 月依次表示为 Jan、Feb、Mar、Apr、May、Jun、Jul、Aug、Sep、Oct、Nov、Dec。日期和年份分别使用阿拉伯数字进行表示，并忽略前导零。例如，Oct, 1, 1949 表示 1949 年 10 月 1 日，Mar, 23, 2024 表示 2024 年 3 月 23 日。

另外，如果生日是不存在的，请输出 **Error!**。如 19198010、20050332、19000229 对应的日期是不存在的。

判断日期是否正确的方法如下：

1. 若月份大于 12，则日期不合法；
2. 若月份为 1,3,5,7,8,10,12，则日期在 1 到 31 之间；
3. 若月份为 4,6,9,11，则日期在 1 到 30 之间；
4. 若月份为 2，则需要判断是否为闰年。满足以下任意条件的即为闰年：
 - (a) 能被 4 整除但不能被 100 整除；
 - (b) 能被 400 整除。

若是闰年，则合法日期在 1 到 29 之间，否则合法日期在 1 到 28 之间。

oql 把这个问题扔给了正在参加新生赛的你。相信你一定可以解决这个问题。

Input

第一行，一个整数 T ($1 \leq T \leq 1000$)，表示身份证号个数。

接下来 T 行，每行一个长度为 18 位的字符串，表示一个身份证号。保证这个身份证号的前 17 位为 0 ~ 9，第 18 位为 0 ~ 9 或 X。不保证地区代号存在和效验码正确。保证年份在 [1900, 2024] 中。

Output

对于每一个身份证，输出一行信息。若身份证中的生日存在，则输出该身份证对应的生日（格式见题目描述）。否则请输出 **Error!**。

请注意，不同身份证的输出间需要用恰好一个换行符隔开；每个英文半角逗号后都需要紧跟一个空格。

Example

standard input	standard output
5	Aug, 10, 1919
114514191908100000	Mar, 23, 2024
202020202403234567	Oct, 1, 2000
000000200010012345	Error!
12345620246543210X	Feb, 20, 2020
202020200220020X	

Problem B. 序列

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

Iris 想到了一个题。
给出一个长度为 n 的序列 a 。定义一个区间 $[l, r]$ 的价值为：

$$\max(a_l, a_{l+1}, \dots, a_r) \times \min(a_l, a_{l+1}, \dots, a_r) \times (r - l + 1)$$

问价值最大的区间价值为多少？
Iris 决定拿这道题考考 oql。

oql 很快便得到了时间复杂度 $O(n)$ 的做法，并决定将这道题放入新生赛。
但他十分好心地缩小了数据范围以便时间复杂度更高的做法能够通过，所以现在请你来尝试解决这个问题。

Input

第一行，一个整数 n ($1 \leq n \leq 10^6$)，表示序列长度。
输入的第二行有 n 个正整数，其中第 i 个数字表示 a_i ($1 \leq a_i \leq 10^9$) 的值。

Output

输出仅一行，包含一个正整数，表示所求的最大区间价值。

Examples

standard input	standard output
10 8 10 7 7 7 5 5 6 5 8	500
10 8 10 4 5 5 4 5 3 4 1	280

Note

请注意答案可能会超过 64 位整数范围。

Problem C. 积分

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

Iris 不喜欢打 codeforces 的比赛，因为比赛通常在 22:35 开始，这不仅导致比赛时容易犯困，而且第二天可能会起不来。

Iris 有两个 codeforces 账号，每次她会根据一些神秘的准则选择使用哪一个账号。

每个账号有自己对应的 rating，初始为 0。因为多种因素的影响，可以认为 Iris 在使用一个账号进行比赛后，新的 rating 为在 $[0, 4000]$ 内随机的一个实数。

那么在 Iris 使用大号进行了 n 场比赛，使用小号进行了 m 场比赛后，大号的历史最大 rating 大于小号的历史最大 rating 的概率是多少。

Input

输入数据共一行，包含两个正整数 n, m ($1 \leq n, m \leq 10^5$)。

Output

输出一个浮点数，表示答案。你的答案将被认为是正确的当且仅当与正确答案的相对误差或绝对误差不超过 10^{-6} 。即，设你输出的答案为 a ，标准答案为 b ，则你的答案是正确的当且仅当 $\frac{|a-b|}{\max(1,b)} \leq 10^{-6}$ 。

Example

standard input	standard output
1 1	0.5000000000

Problem D. 数据包

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

Mandy 要给 brz 连续发送 n 个数据包，但是知周所众，发送数据总是面临着一个无法解决的大难题：数据可能会出错。

这里我们假设一个数据包发送成功的概率为 p ，也就是说有 $1 - p$ 的概率 brz 不能成功接收到这个数据包。但是这些数据包经过了神奇的编码处理，如果 brz 连续接收失败了 k ($k \in [1, n]$) 个数据包，并且紧接着再成功连续接收 k 个数据包，那么接收失败的那 k 个数据包的数据将可以使用神秘算法恢复。

举个例子，用 1 来表示接收成功了一个数据包，0 表示接收失败。则 10011 表示第 1 个数据包接收成功，2,3 数据包接收失败，4,5 数据包接收成功，此时 2,3 数据包被恢复，所以结果是所有数据包成功接收。

但如果是 000110111，也就是在 1,2,3 数据包接收失败后，没有能紧接着接收成功 3 个连续的数据包，那么此时 1,2,3 数据包就无法恢复，彻底接收失败了。需要注意的是，第 7,8,9 个数据包虽然连续成功接收，但是不是紧接着 1,2,3 数据包的，所以无法恢复 1,2,3 数据包。

现在 brz 想要知道，成功接收 Mandy 的这 n 个数据包的概率是多少？

Input

输入一行包含两个整数 n, p' ($1 \leq n \leq 10^6$, $0 \leq p' \leq 100$)，其中 p' 用于表示 $p = \frac{p'}{100}$ 。

Output

输出一行一个整数，表示答案对 998 244 353 取模后的结果。

(我们定义满足 $b \times b^{-1} \equiv 1 \pmod{p}$, $b^{-1} \in [0, p)$ 的整数 b^{-1} 为 b 在模 p 意义下的逆元，则对于一个分数 $\frac{a}{b}$ ，它对 p 取模的值为 $\frac{a}{b} \equiv a \times b^{-1} \pmod{p}$)

Example

standard input	standard output
2 50	499122177

Note

如果是 00，则接收失败。

如果是 01，则用第 2 个数据包可以恢复第 1 个数据包，接收成功。

如果是 10，则接收失败。

如果是 11，则接收成功。

每种情况发生概率均为 $\frac{1}{4}$ ，所以成功接收的概率为 $\frac{1}{2}$ ，对 998 244 353 取模后的结果为 499 122 177。

Problem E. 魔法树场

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

brz 和 Mandy 管理着一个魔法树场，这里面有 n 棵树，排列成一个环的形状，即第 1 棵树的下一棵是第 2 棵树，第 2 棵树的下一棵是第 3 棵树……第 n 棵树的下一棵是第 1 棵树。

第 i 棵树的初始（第 0 天的晚上）高度为 a_i ，这些树很神奇，每棵树在每天白天的时候会生长上一棵树的高度的高度，比如第 1 棵树，过完第一天的白天后，在第一天的夜晚高度会从 a_1 变成 $a_1 + a_n$ 。

而每一天 Mandy 都需要对它们的高度进行一些调查，第 i 天晚上她想知道编号在 $[l_i, r_i]$ 区间内有多少棵树的高度大于 h_i 。

但是 brz 觉得直接统计太麻烦了，他现在想要向你请教：有什么更快的统计方法吗？

当然你不需要向他提供你的方法，他很懒，你只需要告诉他每次统计的结果就好了。

Input

第一行输入两个整数 n, q ($1 \leq n \leq 2 \times 10^5, 1 \leq q \leq 2 \times 10^5$)，表示树的数量和天数。

第二行 n 个整数表示 a_i ($1 \leq a_i \leq 10^{18}$)。

下面 q 行每行三个整数 l_i, r_i, h_i ($1 \leq l_i \leq r_i \leq n, 1 \leq h_i \leq 10^{18}$)。

Output

输出 q 行，每行一个整数表示第 i 天的统计结果。

Example

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

Note

第一天晚上，树的高度依次为 2 3 5 5 3，区间 $[1, 2]$ 内只有 1 棵树高度大于 2。

第二天晚上，树的高度依次为 5 5 8 10 8，区间 $[1, 5]$ 内所有 5 棵树高度都大于 4。

Problem F. 神奇的3

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

0 生 1, 1 生 2, 2 生 3, 3 生万物……Mandy最近发现 3 是一个神奇的数字, 包含着世间的智慧和宇宙中的无穷变化, 因此她想要探究 3 有没有更多的性质。

她找到了一个很大的数字 A , 她想要知道 A 是不是 3 的倍数。但是这个问题太难了, 于是她就丢给了 brz。

brz 向来是很宠 Mandy 的, 但是囿于实力实在不足, brz 也解决不了这个问题。这时 brz 找到了你, 如果你能帮他解决这个问题, 他就会提升你在新生赛的排名作为报酬, 请问你能帮帮他吗?

Input

输入一行一个整数 $A(1 \leq A \leq 10^{1000})$, 表示 Mandy 想要询问的数。

Output

输出一行, 如果 A 是 3 的倍数, 则输出 "Yes" (不含引号), 否则输出 "No" (不含引号)。

Examples

standard input	standard output
1	No
123456789	Yes

Problem G. 世界地图3

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

在 2022 年第 16 届北京邮电大学程序设计竞赛中, oql 出了一道名叫"世界地图"的题目。

在 2023 年北京邮电大学程序设计竞赛新生选拔赛中, oql 出了一道名叫"世界地图2"的题目。

现在, 他要求你继续解决与这个类似的问题!

Iris 正在观看一副世界地图。这个地图是一个 $n \times m$ 的矩阵 A , 行的编号为 $1 \sim n$, 列的编号为 $1 \sim m$ 。 $A_{i,j} = '-'$ 表示点 (i,j) 是水域; $A_{i,j} = '*'$ 表示这是陆地。其中, 若 $A_{i,j}$ 是陆地, 则上面有 $t_{i,j}$ 名游客 (水域上没有游客)。

两个地块 $(x_1, y_1), (x_2, y_2)$ 是相邻的当且仅当 $|x_1 - x_2| + |y_1 - y_2| = 1$ 。

两个陆地地块 X, Y 的最短距离 $\text{dis}(X, Y)$ 定义为: 最小的非负整数, 使得存在满足以下所有条件的地块序列 $b_0, b_1, \dots, b_{\text{dis}(X, Y)}$:

1. $b_0 = X$
2. $b_{\text{dis}(X, Y)} = Y$
3. b_i 均为陆地地块
4. 对于所有的 $i = 0, 1, \dots, (\text{dis}(X, Y) - 1)$, 满足 b_i, b_{i+1} 为相邻地块

若不存在这样地 $\text{dis}(X, Y)$, 则 $\text{dis}(X, Y) = +\infty$ 。特别地, $\text{dis}(X, X) = 0$ 。

在这个世界地图中, 有 k 个旅游景点, 第 i 个景点的人气为 p_i , 坐标为 $S_i = (x_i, y_i)$ 。保证旅游景点的坐标为陆地地块且互不相同。对于任何一个陆地地块 X , 第 i 个旅游景点对这个地块将产生 $\max(0, p_i - \text{dis}(X, S_i))$ 的吸引力。

对于一个陆地地块 (i, j) 上的所有游客 $t_{i,j}$, 如果某个旅游景点 w 对该地块的吸引力都严格大于其他景点对其的吸引力, 则这 $t_{i,j}$ 名游客都将前往该旅游景点进行参观。如果不存在这样的景点 w , 则这 $t_{i,j}$ 名游客哪里都不会去。

Iris 想去旅游, 但并不想去游客太多的景点。Iris 十分好奇每个旅游景点将会有多少游客, 于是她要求 oql 统计出所有景点的游客数量。

oql 不会做, 于是交给了正在参加新生赛的你。

Input

第一行两个正整数 n, m ($1 \leq n, m \leq 500$), 表示地图的大小。

第 $2 \sim (n+1)$ 行, 每行一个长度为 m 的字符串, 表示这个地图中水域与陆地的分布情况。数据保证至少存在一个地块为陆地。

接下来 n 行, 第 i 行 m 个整数 $t_{i,1}, t_{i,2}, \dots, t_{i,m}$ ($0 \leq t_{i,j} \leq 100$), 表示每个地块上的游客数量。保证若地块 (i, j) 为水域, 则 $t_{i,j} = 0$ 。

接下来一行, 一个整数 k ($1 \leq k \leq n \times m$), 表示景点个数。

接下来 k 行, 每行三个整数 p_i, x_i, y_i ($0 \leq p_i \leq 10^6, 1 \leq x_i \leq n, 1 \leq y_i \leq m$), 其中 p_i 表示第 i 个旅游景点的人气, (x_i, y_i) 表示其坐标。保证旅游景点的坐标互不相同。

Output

输出共一行, k 个整数。第 i 个整数表示第 i 个旅游景点最终会有多少的游客。

Examples

standard input	standard output
3 5 *---* ***** *----- 1 0 0 0 1 1 1 1 1 1 1 0 0 0 0 3 3 1 1 3 3 1 5 1 5	1 1 4
3 5 *--*- ***-* *-*** 1 0 0 7 0 1 1 4 0 1 1 0 5 1 4 3 2 3 1 1 1 4 9 2 5	0 7 18
3 5 *---* ***** *----- 1 0 0 0 1 1 1 1 1 1 1 0 0 0 0 3 3 1 1 3 3 1 10 1 5	0 0 8