Problem A. 序列

Input file: `standard input`
Output file: `standard output`

定义函数 $f(x)$，表示序列 x 中不同元素的种类数。例如：

- $f([1,2,3,3]) = 3$，因为该序列包含三种不同的元素：$1, 2, 3$。
- $f([2,2,2,2]) = 1$，因为所有元素相同，仅有一种不同的元素。

进一步地，我们定义函数 $g(y)$，表示序列 y 所有非空子序列 x 对应的 $f(x)$ 之和，即：

$$g(y) = \sum_{x \subseteq y, x \neq \emptyset} f(x)$$

换句话说，$g(y)$ 计算了 y 的所有**非空子序列**中**不同元素种类数**的总和。

一个序列的子序列，指的是从原序列中删除零个或多个元素后，其余元素保持原有相对顺序不变得到的序列。

现在，给定一个正整数 x，你的任务是构造一个序列 a，使得 $g(a) = x$，或报告这样的序列不存在。

Input

输入包含一个正整数 x（$1 \le x \le 10^{18}$）。

Output

如果不存在满足条件的序列，输出一行 `No`。

否则，第一行输出 `Yes`，第二行输出一个整数 n 表示序列长度，第三行输出 n 个正整数 $a_1, a_2, \ldots, a_n$ 表示这个序列。

要求 $1 \le n \le 10^5$，$1 \le a_i \le n$。可以证明如果有解，必然存在满足限制的解。

Examples

standard input	standard output
1	Yes 1 1
4	Yes 2 2 1
10	Yes 3 3 1 3

Problem B. 审判

Input file: `standard input`
Output file: `standard output`

It's a beautiful day outside...

Birds are singing, flowers are blooming...

On days like these kids like you...

SHOULD BE BURNING IN HELL.

你是一个狂妄之人，你来到了最终的审判长廊，接受 sans 的审判。

在一次审判中，sans 会发动数次攻击。攻击分为 k 种类型，其中第 i 类攻击总共会发动 a_i 次（a_i 为非负整数）。在审判开始时，你只知道每种攻击的总次数 $(a_1, a_2, \ldots, a_k)$，但并不知道它们的具体攻击顺序。

你的初始**处决指数 (Execution Points, EP)** 为 1。在 sans 发动每一次攻击前，你可以制定一个应对策略。具体地，在第 j 次攻击前，令你当前的处决指数为 E_j，你可以选择一组实数 $b_1, b_2, \ldots, b_k$，并需满足以下两个条件：

1. $\sum\limits_{i=1}^{k} b_i = 0$
2. 对于**所有** $i \in \{1, 2, \ldots, k\}$，均有 $b_i \geq -E_j$。（无论 sans 发动何种攻击，你的处决指数都不会变为负数）。

随后，sans 会从剩余的攻击中选择一种并发动。若此次攻击为第 l 类，你的处决指数将更新为 $E_j + b_l$。每当一次攻击结束后，你便会确切地知道该次攻击的类型。

sans 会洞悉你的策略，并总是选择最能压制你的攻击顺序，以使你的最终处决指数尽可能低。而你的目标则是制定最优的 b_i 选择策略，来最大化这个由 sans 决定的 “最坏情况” 下的最终处决指数。

这个你通过最优博弈能**保证获得**的最终处决指数，被称为你的**暴力等级 (Level of ViolencE, LV)**。

由于你充满了决心，你将经历 sans 所有可能的审判。一次审判由攻击次数向量 $(a_1, \ldots, a_k)$ 定义。所有可能的审判需满足：

- 总攻击数 $\sum\limits_{i=1}^{k} a_i$ 在 $[0, M]$ 范围内。
- 每种攻击类型的次数 a_i 均不超过 n，即 $0 \leq a_i \leq n$。

两次审判被视为**本质不同**的，当且仅当它们的攻击次数向量 $(a_1, \ldots, a_k)$ 不同。

你是一个狂妄之人，所以你需要求出这若干次审判你的暴力等级之和，答案对 998244353 取模。

Input

一行三个非负整数 $n, k, M (0 \leq n \leq M \leq 10^5, 1 \leq k \leq 10^9)$。

Output

输出一行一个数，表示这若干次审判你的暴力等级之和对 998244353 取模的结果。

Examples

standard input	standard output
1 2 2	7
11 45 14	286390301

Note

对于样例一，你一共会经历 4 次审判。

- 第 1 次审判时，$a_1 = a_2 = 0$，显然你的最终处决指数为 1；
- 第 2 次审判时，$a_1 = 1, a_2 = 0$，你在第 1 次攻击前取 $b_1 = 1, b_2 = -1$，故你的最终处决指数为 2；
- 第 3 次审判时，$a_1 = 0, a_2 = 1$，你在第 1 次攻击前取 $b_1 = -1, b_2 = 1$，故你的最终处决指数为 2；
- 第 4 次审判时，$a_1 = a_2 = 1$，你在第 1 次攻击前取 $b_1 = b_2 = 0$，相当于你跳过一次攻击，此时你得知第 1 次攻击的类型并推断出第 2 次攻击的类型，记为第 i 类，$i \in \{1, 2\}$。从而你在第 2 次攻击前取 $b_i = 1, b_{3-i} = -1$，故你的最终处决指数为 2。

可以证明你按照上述策略所得到的最终处决指数都是能保证得到的最终处决指数中最大的，因此你的暴力等级之和为 $1 + 2 + 2 + 2 = 7$。

Problem C. 饺子

Input file: standard input
Output file: standard output

gsh 非常喜欢吃饺子！每天，他都会前往学苑食堂品尝美味的饺子。食堂每天会供应不同种类和数量的饺子，而 gsh 希望在自己有限的胃容量内，通过合理选择，收获最大的总愉悦值。

食堂今天提供 n 种不同的饺子，gsh 最多能吃 m 个饺子。对于食堂提供的饺子，第 i 种饺子的总数量为 s_i，其基础愉悦值和边际递减系数分别为 a_i 与 b_i。特别地，在首次品尝该种饺子时，会有一个 c_i 的“初见惊喜”加成。

具体来说，吃掉的**第 i 种饺子中的第 j 个**，能获得的愉悦值为 $e_{i,j}$。这个值是预先确定的，与食用的先后顺序无关，计算方式如下：

$$e_{i,j} = \begin{cases} a_i + c_i, & \text{当} j = 1 \text{ 时（即第一次吃第} i \text{ 种饺子）} \\ a_i - b_i \times (j-1), & \text{当} j > 1 \text{ 时} \end{cases}$$

此外，gsh 的食量有一个”完美区间”。如果他吃的饺子总数恰好在 $[l, r]$ 的范围内（包含 l 和 r），他会感到心满意足，从而额外获得 val 点的总愉悦值。

请你帮助 gsh 设计一个吃饺子的方案（即决定每种饺子吃几个），使得他获得的总愉悦值最大化。

Input

第一行输入一个整数 T，表示数据组数（$1 \le T \le 10^5$）。

接下来对每组数据输入如下：

- 第一行输入 5 个整数 $n, m, \text{val}, l, r (1 \le n \le 10^5, \sum n \le 3 \times 10^5, 0 \le m, \text{val} \le 10^6, 0 \le l \le r \le m)$。
- 接下来 n 行，第 i 行输入 4 个整数 s_i, a_i, b_i, c_i（$1 \le s_i, b_i \le 10^6$, $-10^6 \le a_i \le 10^6$, $0 \le c_i \le 10^6$）。

注意：本题不存在对 $\sum m$ 的约束条件。

Output

对每组数据，输出一行一个整数，表示最大愉悦值。

Example

standard input	standard output
3 1 14 5 1 4 19 19 8 10 3 25 40 18 20 20 4 1 4 20 3 1 6 10 -1 2 4 3 25 40 18 20 20 40 3 40 20 30 1 60 10 -10 2 55	48 50 742

Note

对于第一组数据，只有一种饺子。

gsh 吃了 3 个饺子，获得的愉悦值分别是：$29, 11, 3$，同时额外获得 5 点愉悦值（由于饺子总数满足 $l \le 3 \le r$）。

对于第二组数据，有 3 种饺子。

gsh 吃了 8 个第一种饺子，8 个第二种饺子，2 个第三种饺子，获得的愉悦值是 50（包含额外获得的 40 点愉悦值）。

Problem D. 与或博弈

Input file: `standard input`
Output file: `standard output`

gsh 喜欢位运算！今天，他在和一个 AI 进行博弈。

博弈规则如下：

- gsh 和 AI 轮流操作，gsh 先手。
- 他们操作两个非负整数 a 和 b，gsh 的目标是将其变为目标非负整数 x 和 y，而 AI 需要阻止 gsh 达成目标。
- gsh 在自己的回合可以执行以下两种操作之一：
  1. $a := a \,\&\, v$（按位与某个非负整数 v）。
  2. $b := b \mid v$（按位或某个非负整数 v）。
- AI 在自己的回合可以执行以下两种操作之一：
  1. $a := a \mid v$（按位或某个非负整数 v）。
  2. $b := b \,\&\, v$（按位与某个非负整数 v）。
- 允许选择的 v 满足 $0 \le v < 2^{60}$。
- 双方都足够聪明，并且都会采取最优策略以赢得游戏。
- 若在 10^{100} 回合内，存在某一时刻 $a = x$ 且 $b = y$，则 gsh 获胜；否则，AI 获胜。

请你判断 gsh 是否必胜，若必胜，输出 `Yes`，否则输出 `No`。

Input

第一行输入一个整数 T（$1 \le T \le 10^5$），表示数据组数。

接下来对每组数据输入一行四个非负整数 a, b, x, y（$0 \le a, b, x, y < 2^{60}$）。

Output

对每组数据输出一行一个字符串 `Yes` 或 `No`。

Example

standard input	standard output
4	Yes
3 6 3 6	Yes
7 4 5 4	No
5 4 3 4	No
2 4 3 5	

Note

对于第一组数据，初始状态下已经满足 $a = x, b = y$，因此 gsh 必胜。

对于第二组数据，gsh 进行操作 $a := a \,\&\, 5$，此时达到目标，gsh 必胜。

对于第三组数据和第四组数据，不难证明始终无法达到目标，因此 gsh 必败。

Problem E. Django 的数据结构

Input file: `standard input`
Output file: `standard output`

Djangle 很喜欢研究数组操作！

这次，他拿到了一个长度为 n 的正整数序列 a，并希望对它进行一些操作和查询。他对 **gcd（最大公约数）** 的性质十分感兴趣，并提出了如下两种操作，每种操作均包含三个参数 l, r, x：

- **操作 0：区间赋值操作**：将区间 $[l, r]$ 内的所有元素修改为某个给定的正整数 x，即：

$$a_i := x, \quad \forall i \in [l, r]$$

- **操作 1：GCD 查询与更新操作**：
  - 首先，计算区间 $[l, r]$ 内所有元素与某个给定的正整数 x 的 **gcd 之和**，即：

$$\sum_{i=l}^{r} \gcd(a_i, x)$$

  - 然后，将区间 $[l, r]$ 内的所有元素更新为 $\gcd(a_i, x)$，即：

$$a_i := \gcd(a_i, x), \quad \forall i \in [l, r]$$

Djangle 太懒了，他甚至懒得算答案。所以他请你帮助他实现一个程序。

Input

第一行输入一个整数 T（$1 \le T \le 10^5$），表示数据组数。

接下来对每组数据输入如下：

- 第一行包含两个整数 n 和 q（$1 \le n, q \le 10^5$, $\sum n, \sum q \le 10^5$），分别表示数组的长度和操作的次数。
- 第二行包含 n 个正整数 $a_1, a_2, \ldots, a_n$（$1 \le a_i \le 2^{30}$），表示初始数组。
- 接下来的 q 行，每行描述一个操作：
  - 如果是操作 0，格式为 `0 l r x`，表示将区间 $[l, r]$ 内的所有元素修改为 x（$1 \le x \le 2^{30}$）。
  - 如果是操作 1，格式为 `1 l r x`，表示执行 GCD 查询与更新操作（$1 \le x \le 2^{30}$）。

Output

对于每个操作 1，输出计算得到的 gcd 之和。

Example

standard input	standard output
2 5 5 32 16 6 34 47 1 3 4 93 0 2 4 46 0 1 2 81 1 3 5 2 1 3 3 69 10 10 974 560 4 870 975 322 233 742 917 611 1 2 6 766 0 2 6 562 1 4 10 920 1 6 10 353 0 7 10 481 0 9 9 207 1 1 3 43 1 1 10 524 1 1 2 710 0 7 10 190	4 5 1 9 11 5 3 12 2

Note

对于第一组数据:

操作	a_1	a_2	a_3	a_4	a_5	**输出**
初始	32	16	6	34	47	
1 3 4 93	32	16	3	1	47	4
0 2 4 46	32	46	46	46	47	
0 1 2 81	81	81	46	46	47	
1 3 5 2	81	81	2	2	1	5
1 3 3 69	81	81	1	2	1	1

Problem F. No explanation

Input file: standard input
Output file: standard output

给定一个由只包含小写字母的字符串构成的集合 $\{s_i\}$，你需要在集合中选出一些字符串，并选择一个合适的顺序拼接成一个新字符串。设这个新的字符串为 S，出题人**不希望** S 中存在**子序列** `luolikong`，即 S 中**不存在**一个位置集合 $\{\text{pos}_i\}$ 满足

1. 集合的大小为 9 并且 $1 \le \text{pos}_i \le |S|$。
2. 对于所有 $1 \le i \le 8$，$\text{pos}_i < \text{pos}_{i+1}$。
3. 对于长度为 9 并且满足 $t_i = S_{\text{pos}_i}$ 的字符串 t，$t =$ `luolikong`。

~~出题人并不想解释这么做的理由。~~

你只需要计算可能的 S 的最大长度即可。

Input

第一行包含一个整数 n（$1 \le n \le 5000$），表示集合 $\{s_i\}$ 的大小。

接下来的 n 行，每行包含一个字符串 s_i（$1 \le |s_i| \le 5 \times 10^5$，$\sum |s_i| \le 5 \times 10^5$）。

Output

输出一行一个整数，表示可能的 S 的最大长度。

Example

standard input	standard output
5 zhongzhikong luo likelisi kluikong likuong	31

Note

一种合法的拼接方案：zhongzhikong kluikong luo likelisi。

容易证明，不存在另一种拼接方案，使得字符串长度大于 31 且不含子序列 luolikong。

Problem G. 矩阵

Input file: `standard input`
Output file: `standard output`

给定一个正整数 n，你需要构造一个 $n \times n$ 的矩阵。

矩阵的第 i 行第 j 个元素，记作 $A_{i,j}$。

你的目标是使得：对于所有 $1 \le i \le n$ 且 $1 \le j \le n$，都有

- 如果 $i > 1$，有 $\gcd(A_{i,j}, A_{i-1,j}) = 1$
- 如果 $j > 1$，有 $\gcd(A_{i,j}, A_{i,j-1}) = 1$

并且所有的数字都满足 $1 \le A_{i,j} \le n^2 + 40n$，**且所有数字互不相同**。

也就是说，矩阵中，所有的数字与它上下左右相邻的四个数字都互质，并且矩阵中的数都不超过 $n^2 + 40n$。

$\gcd(x, y)$ 表示 x 和 y 的最大公约数。

Input

一行一个正整数 $n(1 \le n \le 2500)$。

Output

输出 n 行，每行 n 个正整数，用空格分隔。如果有多种满足要求的解，输出任意合法解即可。

Examples

standard input	standard output
2	1 3 5 4
3	1 5 9 4 7 8 3 2 11

Note

本题输出数据量较大，请在程序开头添加以下语句以关闭输入输出流同步，加快输出速度：

```
ios::sync_with_stdio(false);
cin.tie(nullptr);
cout.tie(nullptr);
```

Problem H. V 我 112.5

Input file: `standard input`
Output file: `standard output`

很遗憾通知大家，受美国 125% 关税影响，今天的 v 我 50 活动取消，改为 v 我 112.5。

每逢周四，关税政策的变动都会影响需要支付的费用。给定当前关税百分比 x，计算你需要支付的总金额。费用计算规则如下：

- 基础费用固定为 50 元
- 额外费用 $=$ 基础费用 $\times$ 关税百分比 $x\%$
- 总费用 $=$ 基础费用 $+$ 额外费用

Input

一个整数 x（$0 \le x \le 19198100$），表示关税税率为 $x\%$。

Output

输出总费用，保留三位小数（四舍五入），格式为：Vivo [金额]

Examples

standard input	standard output
100	Vivo 100.000
125	Vivo 112.500
735568	Vivo 367834.000
0	Vivo 50.000

Problem I. 真相

Input file: standard input
Output file: standard output

有一棵包含 n 个节点的有根树，**其根节点为 1**。每个节点上站着一个人，每个人要么是“诚实者”（总是说真话），要么是“说谎者”（总是说假话）。

现在，对于每个 $i \in [1, n]$，位于节点 i 的人都说了一句话：“以我所在的节点为根的子树中，恰好有 a_i 个诚实者。”

一个“真假分配情况”是指为每个节点的人分配一个“诚实者”或“说谎者”的身份。你需要计算，有多少种可能的真假分配情况，能够满足以下**逻辑自洽条件**：

- 对于任意一个节点 i，如果该节点的人是**诚实者**，那么他陈述的数字 a_i 必须**等于**其子树中诚实者的实际总数。
- 对于任意一个节点 i，如果该节点的人是**说谎者**，那么他陈述的数字 a_i 必须**不等于**其子树中诚实者的实际总数。

gsh 对真相非常感兴趣，他想知道满足上述条件的真假分配情况总共有多少种。请输出答案对 998244353 取模的结果。

Input

第一行输入一个整数 T（$1 \le T \le 5000$），表示数据组数。

接下来对每组数据输入如下：

- 第一行输入一个正整数 n（$1 \le n \le 5000$，$\sum n \le 5000$），表示节点数量。
- 第二行输入 n 个以空格分开的非负整数 a_i（$0 \le a_i \le n$）。
- 接下来的 $n-1$ 行，每行输入两个正整数 u_i，$v_i (1 \le u_i, v_i \le n)$，表示一条边。保证输入的边构成一棵树。

Output

对每组数据输出一行一个整数表示答案对 998244353 取模后的结果。

Example

standard input	standard output
6 3 1 2 3 1 2 1 3 3 0 2 3 1 2 1 3 3 0 2 1 1 2 1 3 5 5 1 1 1 4 5 1 4 5 5 3 2 5 5 5 1 3 4 1 5 3 2 3 1 4 4 3 6 3 5 1 0 1 0 2 6 4 5 2 3 6 1 4 2	2 0 1 10 7 3

Note

对于第一组数据，有 2 种可能：

1. 1 号说真话，2 号说假话，3 号说假话。
2. 1 号说假话，2 号说假话，3 号说假话。

对于第二组数据，没有合法方案。

对于第三组数据，只有唯一一种可能：1 号说假话，2 号说假话，3 号说真话。

Problem J. 画圈

Input file: `standard input`
Output file: `standard output`

Djangle 给你一个连通的简单无向图，初始时每条边都有一个颜色：**白色**或**黑色**。

每次操作，你可以选择一个**包含至少一条白边**的**简单环**，将这个环中所有的边都涂成**黑色**。**请注意，你并不一定需要最终把所有边都变为黑色**。

请问，最多可以进行多少次这样的操作？

简单环的定义是：由若干条边首尾连接而成的闭合路径，且其中没有重复的边。

Input

第一行包含一个整数 $T(1 \le T \le 10^4)$，表示数据组数。

接下来每组数据的第一行包含两个整数 $n, m(1 \le n \le \sum n \le 2\times10^5, n-1 \le m \le \min\{3\times10^5, \frac{n\times(n-1)}{2}\}$, $\sum m \le 3 \times 10^5)$，表示图的点数和边数。

接下来 m 行，每行包含三个整数 $u, v, \mathrm{col}(1 \le u, v \le n, \mathrm{col} \in \{0, 1\})$，表示存在一条连接点 u 和点 v 的无向边，颜色为 col。如果 $\mathrm{col} = 0$，表示该边为白色；否则表示该边为黑色。

保证输入的图没有重边和自环。

Output

对于每组数据，输出一个整数，表示最多可以进行多少次操作。

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>1
9 10
1 2 1
2 3 0
3 4 1
4 5 0
5 6 1
6 7 0
7 8 1
8 1 0
2 9 0
9 6 1</td><td>2</td></tr>
</table>

Note

第一次操作简单环 $2-9-6-7-8-1-2$，第二次操作简单环 $1-2-3-4-5-6-7-8-1$。

容易证明，不存在其他的操作方案，能够操作 3 次以上。

Problem K. 神之一手

Input file: standard input
Output file: standard output

“本手、妙手、俗手、举手”是围棋的四个术语。**本手**是指合乎棋理的正规下法；**妙手**是指出人意料的精妙下法；**俗手**是指貌似合理，而从全局看通常会受损的下法。棋手应该从本手开始，本手的功夫扎实了，棋力才会提高，才能下出妙手，否则难免下出俗手。不过在你主场作战又下不出妙手时，你也可以尝试通过**举手**直接获得胜利。

近日 H 国举行了围棋比赛邀请您参加，您一路过关斩将。为增加比赛难度，H 国特推出一条新规则。棋手提子后，必须放入棋盒盖中。若有两枚或更多棋子没有被成功放入棋盒盖中，你会立刻输掉这盘棋。这些没能成功留在棋盒盖中的棋子，我们称之为**界外棋子**。

您马上要和 H 国选手进行决赛第二轮的比拼，因为对方是 H 国本国选手，所以他可以使用举手的方式额外增加比赛难度。

形式化地，我们可以将问题简化成如下的情景：

行棋总共有 n 回合，每回合都严格顺次进行如下阶段。在任何一个阶段中，一旦您的界外棋子总数达到 2，或被对方成功举手，您都将**立刻**输掉比赛。

1. 回合开始时，若您恰有 1 颗界外棋子，H 国选手有 $\frac{r}{1000}$ 的概率举手，使您直接败北。
2. 之后，您将获得一堆数量为 a_i 的棋子，您会从这堆棋子中逐个拿出棋子，并将其放入棋盒盖中，直至这堆棋子为空。由于您思考时过于专注，每颗棋子有 $\frac{p}{1000}$ 的概率失误而**不放入**棋盒盖，直接成为界外棋子。
3. 在放入一颗棋子时，若你之前**拿出**的棋子总数大于或等于 80（**无论最终有没有放入棋盒盖中**），那么这枚刚被放入的棋子将有 $\frac{q}{1000}$ 的概率从过满的棋盒盖中掉出，同样成为一颗界外棋子。在上述过程中，只要您的界外棋子总数累计达到 2，您就立刻败北。

因为决赛中您思考十分专注，并且对此规则并不熟悉，所以您并不会注意到您掉出棋盒盖的棋子，也不会将它们放回。

现在，对于 n 个回合中的每一个回合，请计算您恰好在此回合因为规则或对方举手而输掉的概率。

Input

第一行包含三个非负整数 p，q，r（$0 \le p, q, r \le 1000$），表示各项规则的概率参数。

第二行包含一个正整数 n（$1 \le n \le 10^5$），表示比赛回合数。

第三行包含 n 个非负整数 $a_1, a_2, \ldots, a_n$（$0 \le a_i \le 10^9$），表示每回合的提子数。

Output

输出 n 行，每行表示当回合的失败概率。若概率为 0 则输出 0，否则输出 $PQ^{-1} \bmod (10^9+7)$，其中 $\frac{P}{Q}$ 为最简分数形式。

可以证明，每回合失败概率要么是 0，要么可以表示为 $\frac{P}{Q}$（P，Q 互质）。

Examples

standard input	standard output
0 500 1000 3 79 2 0	0 0 500000004
0 500 0 2 79 3	0 250000002
500 500 500 3 1 1 1	0 375000003 531250004
431 404 519 20 0 8 10 8 10 0 8 9 10 2 0 6 0 7 4 1 5 3 6 5	0 65193629 491317294 880422337 670795490 121913119 186949131 942576908 924311201 247154371 759862879 967726799 679692511 202922928 490250649 551050329 827900501 946464281 916292188 461482115

Note

对于样例一，$p' = 0, q' = \frac{1}{2}, r' = 1$。在第一回合，您提了 79 子，全部成功放入棋盒盖。在第二回合，您提的第一颗棋子成功放入棋盒盖，第二颗棋子有 $\frac{1}{2}$ 的概率没能放入棋盒盖。在这一情形下，对方在第三回合一定会举手，因此您在第三回合输掉的概率为 $\frac{1}{2}$。

对于样例二，$p' = 0, q' = \frac{1}{2}, r' = 0$。在第一回合，您提了 79 子，全部成功放入棋盒盖。在第二回合，您提的第一颗棋子成功放入棋盒盖，后两颗棋子各自独立地有 $\frac{1}{2}$ 的概率没能放入棋盒盖。因此，您在第二回合输掉的概率为 $\frac{1}{4}$。

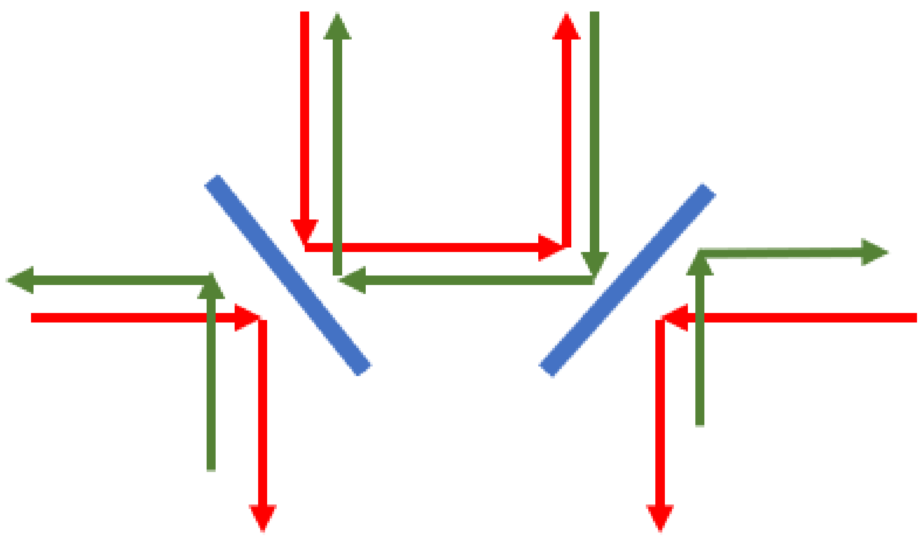
对于样例三，您在第二回合输掉的概率是 $\frac{3}{8}$，在第三回合输掉的概率是 $\frac{9}{32}$。

Problem L. 迷宫

Input file: standard input
Output file: standard output

jzh 有一个 $n \times m$ 的迷宫，每个格子上都有一面反弹板。反弹板的类型可能是以下两种之一：

1. / （右斜板）：
   - 小球从左侧射入 → 反弹向上方
   - 小球从右侧射入 → 反弹向下方
   - 小球从上方射入 → 反弹向左侧
   - 小球从下方射入 → 反弹向右侧
2. \ （左斜板）：
   - 小球从左侧射入 → 小球向下方
   - 小球从右侧射入 → 小球向上方
   - 小球从上方射入 → 小球向右侧
   - 小球从下方射入 → 小球向左侧



迷宫被矩形反弹板包围，如果小球垂直撞击迷宫边界上的反弹板，小球会原路返回。

Djangle 有两个锤子，他可以在小球运动过程中**任意时刻**锤碎最多两个反弹板（**可以先用反弹板反弹后锤碎反弹板**），小球经过被锤碎的反弹板时不会发生反弹。

Djangle 锤碎位于第 i 行第 j 列的反弹板需要耗费 $a_{i,j}$ 的体力。

注意：Djangle 可以使用两个锤子、一个锤子或者不使用锤子。

gsh 觉得这个过程很有趣，他想知道：小球从一个格子按照某一方向发射，能否到达另一个格子，如果能，Djangle 最少要花费多少体力。

Input

第一行三个整数 n，m，q（$1 \le n, m \le 5 \times 10^5$，$1 \le n \times m \le 4 \times 10^6$，$1 \le q \le 10^6$）分别表示迷宫的行，列，和询问次数。

接下来 n 行，第 i 行第 j 个字符 $S_{i,j} \in \{\backslash, /\}$ 表示第 i 行第 j 列的反弹板样式。

接下来 n 行，每行包含 m 个正整数，表示 $a_{i,j}$（$1 \le a_{i,j} \le 10^9$）。

接下来 q 行，每行包含用空格分开的五个询问参数：

起始坐标：$(\text{startx}, \text{starty})$。

发射方向：$\text{direction} \in \{\texttt{N}, \texttt{S}, \texttt{W}, \texttt{E}\}$ 分别代表北（上），南（下），西（左），东（右）。

目标坐标：$(\text{endx}, \text{endy})$。

$1 \le \text{startx}, \text{endx} \le n, 1 \le \text{starty}, \text{endy} \le m$。

Output

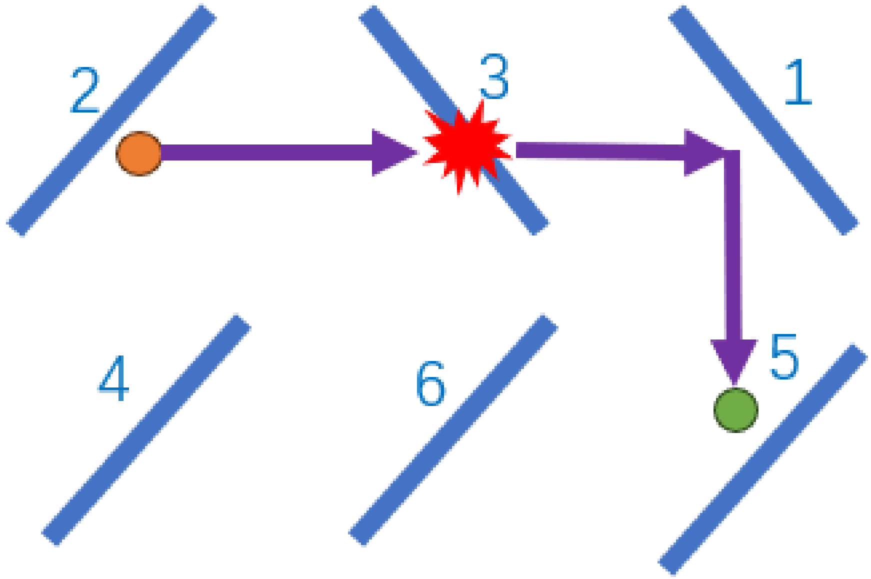
对于每一个询问，输出一个整数，表示 Djangle 最少需要花费多少体力。如果小球无论如何都不能到达目标格子，输出 -1。

Examples

standard input	standard output
2 3 1 /\\ /// 2 3 1 4 6 5 1 1 E 2 3	3
3 3 5 \\\ \/\ /\\ 3 1 1 1 4 1 2 5 1 1 3 N 1 2 3 3 S 2 3 2 3 W 2 3 3 1 S 1 2 2 2 S 1 2	1 0 0 3 1

Note

对于样例一：


2
3
1
4
6
5

Problem M. 魔法使考核

Input file: standard input
Output file: standard output

见习魔法少女 Chiaro 正在参加魔法使考试，大魔法使 Shiro 在她面前放置了 n 个没有任何魔力值的魔法球，Chiaro 需要施加若干次魔法，让这些魔力球各装满指定大小的魔力值。

具体来说，初始时这些魔法球的魔力值都是 0，Chiaro 要在施加若干次魔法后使得第 i 个魔法球最终有 a_i 的魔力值。然而 Chiaro 作为见习魔法少女掌握的魔力知识很少，她只会两种魔法，而且每种魔法都会消耗一定的体力：

- 将任意一个魔法球增加 1 的魔力值，这会消耗她 x 点体力。
- 选择任意区间 $[l, r]$，将第 l 个到第 r 个共 $r - l + 1$ 个魔法球的魔力值翻倍，这会消耗她 y 点体力。

很显然，Chiaro 一定可以只用上述两种魔法若干次来完成这项测试。不过这场魔法使考试才刚刚开始，后面还有很多更困难的测试项目，所以 Chiaro 想用尽可能少的体力完成这个测试，请你帮她算出最少用多少体力可以完成这个测试。

Input

第一行三个非负整数 $n, x, y (1 \le n \le 3 \times 10^5, 0 \le x, y \le 10^7)$。

接下来一行 n 个非负整数表示序列 $a_1, a_2, \cdots, a_n (0 \le a_i \le 10^9)$。

Output

一行一个整数，表示消耗的最少体力值。

Examples

standard input	standard output
6 1 1 1 1 4 5 1 4	9
5 2 5 10 7 9 0 3	32