

A. 赛马比赛

题目描述

愉快的寒假开始了，你终于有时间来看你最喜欢的赛马比赛了！

赛马的规则是这样的：

每一匹马都有一个起始位置 p_i 和一个奔跑速度 v_i 。

对于一匹赛马，如果在奔跑过程中被在其后的赛马超过（在同一位置不算超过），它就会摆烂而退出比赛。

由于你有密集恐惧症，所以如果想要舒服地观看的话，正在比赛的赛马的数量不能超过 k 个，请问你至少需要等待多久才能一直舒适的观看比赛。

输入格式

第一行两个整数 n, k (初始的赛马的数量，最多可以出现的赛马的数量)

第二行 n 个整数 p_1, p_2, \dots, p_n (赛马的初始位置)

第三行 n 个整数 v_1, v_2, \dots, v_n (赛马的奔跑速度)

$$1 \leq n \leq 10^5$$

$$1 \leq k \leq n$$

$$1 \leq p_i \leq 10^9$$

$$1 \leq v_i \leq 10^9$$

数据保证最初不会有二个赛马位于同一个位置

输出格式

如果可以舒适的观看接下来的比赛的话，请输出至少需要等待的时间 否则输出Never!

样例 #1

样例输入 #1

```
3 2
1 3 4
2 4 2
```

样例输出 #1

```
1
```

样例 #2

样例输入 #2

```
3 2
1 3 4
2 2 2
```

样例输出 #2

```
Never!
```

B.数字顶真

题目描述

一些文化中，7 象征着好运、成功和幸福，在中国文化中，7 与“起”谐音，象征着事业的起步和发展，也被视为一个幸运数字。因此如果一个数字它其中的一位包含 7，或者这个数字可以被 7 整除，我们称这个数为天真的数。

现在，小明想知道全部大于零的天真的数，问你第 k 个天真的数是什么

输入格式

第一行一个整数 t ，表示数据组数。

接下来的 t 行，每行一个整数 k 。

输出格式

一共 t 行，第 i 行输出第 i 组数据的答案。

样例 #1

样例输入 #1

```
11
1
2
3
4
5
6
7
8
9
10
2021
```

样例输出 #1

7
14
17
21
27
28
35
37
42
47
5477

提示

对于 100% 的数据, $1 \leq t, k \leq 2024$ 。

C. 2bit

题目描述

某一天, 小 q 给了你一个神奇的二进制数, 你只需要花 1 元的代价, 就可以交换二进制数相邻的两位

于是小 q 准备考一考你, 他给了你 n 个问题, 对于第 i 个问题:

如果你要把这个二进制数变成 2^i 的整数倍? 最少需要花费多少钱, 如果不可能, 则输出 -1 。

你能解决这个问题吗?

输入格式

第一行, 一个整数 n 。 $1 \leq n \leq 100000$ 。

第二行, n 位二进制数, 每位是 0 或 1。

输出格式

一行, 共 n 个整数。

输入:

```
5
10101
```

输出:

```
1 3 -1 -1 -1
```

D. 黑白格子

题目描述

总所周知，国际象棋的棋盘是由黑白相间的格子构成的，现在给你一个 N 行 M 列的格子棋盘，每个格子都要染成黑色或者白色。

如果格子棋盘每个相邻的格子之间的颜色都互不相同，那么这个棋盘则是合格的棋盘

现在，你可以把一种颜色的格子变成另外一种颜色，代价为 1

如果把一个棋盘变成合格的棋盘，最少要花费 P 的代价，则这个棋盘的代价是 P 。

下面的棋盘的代价就是 2，至少通过两次变换才能变成合格的棋盘

```
WBWBW
BWBWB
BBWWW
BWBWB
```

容易发现，代价是 P 的棋盘，可能会有很多种。

求：总共有多少种不同的代价是 P 的“普通”棋盘？。

输入格式

一行，3 个整数， N, M, P 。 $1 \leq N, M \leq 100, 0 \leq P \leq 100$ 。

输出格式

一个整数表示答案，输出结果对 $10^9 + 7$ 取模。

输入/输出例子1

输入：

```
2 2 1
```

输出:

8

输入/输出例子2

输入:

9 4 15

输出:

135805043

E.Mex

题目描述

我们定义数组的 *Mex* 是不属于这个数组的最小非负整数。示例：

对于数组 $[0, 0, 1, 0, 2]$, *Mex* 等于 3, 因为数字 0, 1 和 2 出现在数组中, 3 是数组中未出现的最小非负整数;

对于数组 $[1, 2, 3, 4]$, *Mex* 等于 0, 因为 0 是数组中未呈现的最小非负整数;

对于数组 $[0, 1, 4, 3]$, *Mex* 等于 2, 因为 2 是数组中未出现的最小非负整数。

现在给您一个空数组 $a = []$ (换句话说, 一个零长度数组) 。和一个正整数 x 。

还将向您提供 q 个操作。在每次操作中, 我们会往数组里加入一个数, 数组长度增加 1。

然后, 您可以选择任意一个元素 a_i , 并对 a_i 执行任意次 $a_i = a_i - x$ 或者 $a_i = a_i + x$ 操作, 唯一的限制是 a_i 不能变成负数。

现在让您尽可能使每次操作后 *Mex* 最大

输入描述

输入的第一行包含两个整数 q, x ($1 \leq q, x \leq 4 \cdot 10^5$) - 查询的数量和 x 的值。

接下来的 q 行描述查询。第 j 个查询由一个整数 y ($0 \leq y \leq 10^9$) , 这意味着要将一个元素 y 添加到数组中。

输出描述

在每次查询后打印初始问题的答案-对于查询 j , 在第一次 j 次查询后打印 *Mex* 的最大值。请注意, 查询是相关的 (每次查询后数组都会更改) , 但查询之间的操作是独立的。

输入输出

输入#1

```
7 3
0
1
2
2
0
0
10
```

输出#1

```
1
2
3
3
4
4
7
```

输入#2

```
4 3
1
2
1
2
```

输出#2

```
0
0
0
0
```

提示

在第一个示例中：

在第一次查询之后，数组是 $a = [0]$ ：您不需要执行任何操作，最大可能的Mex是1

在第二次查询之后，数组是 $a = [0, 1]$ ：您不需要执行任何操作，最大可能的Mex是2。

在第三次查询之后, 数组是 $a = [0, 1, 2]$: 您不需要执行任何操作, 最大可能的Mex是3。

在第四个查询之后, 数组是 $a = [0, 1, 2, 2]$: 您不需要执行任何操作, 最大可能的Mex是3 (不能通过操作使其更大)。

在第五次查询后, 数组为 $a = [0, 1, 2, 2, 0]$: 可以执行 $a[4] = a[4] + 3 = 3$ 。数组变为 $a = [0, 1, 2, 2, 3]$ 。现在Mex是可能的最大值, 等于4。

在第六次查询之后, 数组为 $a = [0, 1, 2, 2, 0, 0]$: 您可以执行 $a[4] = a[4] + 3 = 0 + 3 = 3$ 。数组变为 $a = [0, 1, 2, 2, 3, 0]$ 。现在Mex是可能的最大值, 等于4。

在第七次查询之后, 数组是 $a = [0, 1, 2, 2, 0, 0, 10]$ 。您可以执行以下操作:

$$a[3] = a[3] + 3 = 2 + 3 = 5,$$

$$a[4] = a[4] + 3 = 0 + 3 = 3,$$

$$a[5] = a[5] + 3 = 0 + 3 = 3,$$

$$a[5] = a[5] + 3 = 3 + 3 = 6,$$

$$a[6] = a[6] - 3 = 10 - 3 = 7,$$

$$a[6] = a[6] - 3 = 7 - 3 = 4.$$

结果数组将为 $a = [0, 1, 2, 5, 3, 6, 4]$ 。现在Mex是最大值, 等于7。

F.干草

题目描述

有 n 袋干草，第 i 袋干草的重量是 $w[i]$ 。奶牛当前的快乐值是 0，你希望它的快乐值至少要达到 s 。

如果奶牛吃掉第 i 袋干草，奶牛的快乐值会增加 $w[i]$ 。

对于一袋干草来说，奶牛要么整袋吃掉，要么不吃，不能吃这袋干草的一部分。

如果奶牛当前的快乐值小于 s ，那么奶牛必须要继续挑选干草吃。

奶牛最近的感知功能不是很好，有一个延迟参数 t 。

如果 t 等于 0，那么奶牛当前快乐值只要不小于 s ，那么它就不再吃干草了。

如果 t 是正整数，那么奶牛快乐值达到 s 以后，仍然要额外多吃 t 袋干草。

求奶牛能吃到的干草的总重量的最大值是多少。

注意：有可能吃完所有的干草后，快乐值仍然没达到 s 。

输入格式

第一行，三个整数： n, s, t 。 $1 \leq n \leq 100, 1 \leq s \leq 10000, 0 \leq t \leq 100$ 。

第二行， n 个正整数，第 i 个整数是 $w[i]$ 。所有 $w[i]$ 的总和不超过 1000000000。

输出格式

一个整数。

输入/输出例子1

输入：

```
4 1234 0
10 20 30 40
```

输出:

```
100
```

输入/输出例子2

输入:

```
3 100 0  
100 100 100
```

输出:

```
100
```

输入/输出例子3

输入:

```
5 101 2  
100 100 100 100 100
```

输出:

```
400
```

G.falcon

题目描述

定义 $C_{i,j}$

$$C_{i,j} = \begin{cases} \max(i, j) & i = 1 \vee j = 1 \\ \frac{C_{i-1,j} + C_{i,j-1} + i + j}{2} & \text{otherwise} \end{cases}$$

求 $\sum_{i=1}^n \sum_{j=1}^m C_{i,j}$

输入格式

输入两个整数 $n, m (1 \leq n, m \leq 10^{18})$;

输出格式

一行一个数表示答案，对 1000000007 取模。

样例输入

```
3 3
```

样例输出

```
36
```

H.工大战士

题目描述

你说的对，但是《原鱼》是由 QLU-ACM 协会自主研发的一款全新开放世界冒险游戏，考虑到游戏已经家喻户晓，所以在此省略两万字的游戏介绍。正值春节版本，游戏中推出了一名新的翔元素双手键角色“QLU战士”，该角色一共有三个技能：

1. 普通攻击【翔龙在天】：一共分为 n 段，使用第段普通攻击会花费 t_i 秒的时间，造成 d_i 点伤害，并让角色获得点怒气值。只有连续使用普通攻击时才会叠加段数，第 n 段普通攻击结束后或者使用了其他技能后再使用普通攻击，都会从第 1 段开始**重新**计算。
2. 怒气爆发【翔空万里】：当怒气值达到 E 时才能使用，使用后怒气值将变为 0。使用怒气爆发会花费 t_a 秒的时间，造成 d_a 点伤害，并让角色获得增益效果之后的 m 次**普通攻击造成的伤害翻倍**。如果在增益效果结束前再次使用怒气爆发，会**清除**之前的增益效果并**重新获得**增益效果。怒气爆发**使用完毕后**该技能的冷却时间将变成 u_a 秒。
3. 怒气战技【龙翔虎跃】：使用怒气战技会花费 t_b 秒的时间，造成 d_b 点伤害，让角色获得 c_b 点怒气值并减少怒气爆发 w 秒的冷却时间。怒气战技**使用完毕后**该技能的冷却时间将变成 u_b 秒。

友情提示：每个技能的冷却时间都是**独立**计算的，每秒钟所有技能的冷却时间都会减少 1 秒，冷却时间最少为 0 秒，当技能的冷却时间**大于** 0 秒时不能使用该技能。任意技能在使用过程中都不会被打断，每个技能在使用完**毕**后才能使用下一个技能。

QLU-ACM协会的游戏策划现在需要测试一下新角色的强度，但由于他很懒，所以他只会按照最简单的方式使用技能：可以使用技能时，优先使用怒气爆发，其次使用怒气战技，最后使用普通攻击。

初始时角色的怒气值为 0，怒气战技和怒气爆发的冷却时间也均为 0，没有增益效果也没有叠加普通攻击的段数。现在策划想知道，如果想要造成 s 点伤害，至少需要多少秒？为了历练新人，所以策划将任务分配给了新招的临时工，也就是你。

输入格式

第一行包含两个整数 n 和 s ，表示普通攻击分为 n 段，总共需要造成 s 点伤害。

第二行包含五个整数 E 、 m 、 t_a 、 d_a 和 u_a ，表示怒气值达到 E 后才能使用怒气爆发，获得 m 次普通攻击的增益，花费 t_a 秒时间，造成 d_a 点伤害，冷却时间变成 u_a 秒。

第三行包含五个整数 w 、 e_b 、 t_b 、 d_b 、 u_b ，表示怒气战技减少怒气爆发 w 秒冷却，获得 e_b 点怒气值，花费 t_b 秒时间，造成 d_b 点伤害，冷却时间变成 u_b 秒。

接下来 n 行每行包含三个整数 t_i 、 d_i 和 e_i ，按顺序描述每一段普通攻击，花费 t_i 秒时间，造成 d_i 点伤害，获得 e_i 点怒气值。

输出格式

输出一个整数，表示要造成 s 伤害需要的最少时间。

数据样例

【样例 1 输入】

```
2 22
1 0 1 1 1
3 0 3 4 7
2 3 0
1 2 0
```

【样例 1 输出】

```
16
```

【样例 1 解释】

由于所有技能获得的怒气值均为 0，所以一直无法使用怒气爆发。

首先使用怒气战技，花费 3 秒，造成 4 点伤害，该技能冷却时间变成 7 秒。

然后依次使用普通攻击第 1, 2, 1, 2, 1 段，共花费 8 秒，造成 13 点伤害。

此时怒气战技的冷却时间为 0 秒，再次释放怒气战技，花费 3 秒，造成 4 点伤害。

最后使用普通攻击第 1 段，花费 2 秒，造成 3 点伤害。

此时总伤害为 24，达到目标，共花费 16 秒。

【样例 2 输入】

```
6 2 1 1 10
3 3 1 2 5
2 3 4
1 2 5
```

【样例 2 输出】

13

【样例 2 解释】首先使用怒气战技，花费 1 秒，造成 2 点伤害，获得 3 点怒气值，该技能冷却时间变成 5 秒。

然后使用普通攻击第 1 段，花费 2 秒，造成 3 点伤害，获得 4 点怒气值。

此时怒气值达到 7，使用怒气爆发，怒气值变成 0，花费 1 秒，造成 1 点伤害，获得 2 次普通攻击的增益，该技能冷却时间变成 10 秒。

然后使用增益普通攻击第 1 段，花费 2 秒，造成 6 点伤害，获得 4 点怒气值。

此时怒气战技的冷却时间为 0 秒，再次释放怒气战技，花费 1 秒，造成 2 点伤害，获得 3 点怒气值，并减少了怒气爆发 3 秒的冷却。

然后使用增益普通攻击第 1 段，普通攻击第 2, 1 段，共花费 5 秒，造成 11 点伤害，获得 13 点怒气值。

此时怒气战技和怒气爆发的冷却时间均为 0，怒气值达到 20，优先使用怒气爆发，花费 1 秒，造成 1 点伤害。

此时总伤害为 26，达到目标，共花费 13 秒。

数据说明

对于所有测试点，满足：关于普通攻击段数、增益效果与伤害总量： $1 \leq n \leq 5, 0 \leq m \leq 10, 1 \leq s \leq 10^{18}$ 关于技能的耗时、伤害与怒气： $1 \leq t_a, t_b, t_i, d_a, d_b, d_i \leq 3, 0 \leq e_i, e_b, w \leq 3$ 关于技能的冷却时间与怒气条件： $0 \leq u_a \leq 30, 0 \leq u_b \leq 15, 1 \leq E \leq 80$ 。

I.Mex重组

题目描述

我们定义数组的 Mex 是不属于这个数组的最小非负整数。示例：

对于数组 $[0, 0, 1, 0, 2]$, Mex 等于 3, 因为数字 0, 1 和 2 出现在数组中, 3 是数组中未出现的最小非负整数;

对于数组 $[1, 2, 3, 4]$, Mex 等于 0, 因为 0 是数组中未呈现的最小非负整数;

对于数组 $[0, 1, 4, 3]$, Mex 等于 2, 因为 2 是数组中未出现的最小非负整数。

现在给你一个包含 n 个元素的集合, 你要把它重组为若干个新的集合, 让这些集合的 Mex 和最大, 请求出这个最大值是多少

输入格式

第一行输入一行一个正整数 n , 接下来一行包含 n 个非负整数 a_i

输出格式

输出一个整数表示答案。

样例 #1

样例输入 #1

```
5
0 0 1 1 2
```

样例输出 #1

```
5
```

数据范围

对于所有数据, 有 $1 \leq n \leq 10^5, 0 \leq a_i \leq 1000$

样例解释

对于例1, 分为两个集合 $\{0, 1, 2\}$ 和 $\{0, 1\}$, *Mex* 为3和2, 所以答案为 5

J.攻击的车

题目描述

国际象棋的棋盘可以分为 r 行 c 列。棋盘上放置了 n 只车。第 i 只车在第 x_i 行, 第 y_i 列。每只车可以控制同一行及同一列的方格 (包括自身)。请统计这些车一共可以控制多少个方格。

输入格式

第一行: 三个整数: 表示 r, c 与 n

第二行到第 $n + 1$ 行: 第 $i + 1$ 行有两个整数表示 x_i 与 y_i 。

输出格式

单个整数: 表示答案。

样例

```
2 3 1
1 2
```

```
4
```

说明

RRR

R

数据范围

- 100% 的数据, $1 \leq r, c \leq 10^9$
- $1 \leq n \leq 10^6$
- $1 \leq x_i \leq r$

- $1 \leq y_i \leq c$
- 保证同一个方格不会出现多只车。