POTYCZKI ALGORYTMICZNE

问题A AGI

第三届宇宙杯，第40阶段：Potyczki。限制：1024 MB，2秒。 *2025年6月21日*

通用人工智能（简称AGI）似乎日益不可避免。越来越多的人不再质疑我们能否达到这一水平，而是*关注何时实现*。迄今已有*众多*未来学家提出关于AGI出现时间的预测。第i项预测以时间区间[A_i，B_i]形式给出，这意味着

≤该预测认为AGI将在*满足*$A_i \leq t$ 且 $t < B_i$的时间点t出现。

对于每项预测，你必须决定是否认为其为真。

你的任务是做出这些判断，确保无论通用人工智能最终出现的实际*时刻x*为何时，都能保证至少有(n)-1个正确评估。

可假设测试用例的选取方式保证至少存在一2个有效答案。需解决t个独立测试用例。

输入

第一行包含一个整数 t（$1 \leq t \leq 1000$）—— 表示测试用例的数量。每个测试用例包含一行整数 n（$1 \leq n \leq 500\,000$），表示预测数量。随后 n 行描述预测内容，每行包含两个整数 A_i和 B_i（$0 \leq A_i < B_i \leq 10^9$），分别表示第 i 次预测的时间区间的起始和结束时间。

所有测试用例中n的总和不超过500 000。

输出

对于每个测试用例，输出包含字符串T和N的长度*为n的*字符串。该字符串的第j个字符是你对第j个预测的评估：

- T 表示你确认该预测。
- N 表示你拒绝该预测。

若存在多个满足题目的答案，可输出任意一个。

示例

输入示例：

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

正确的输出示例如下：

```
NNNN
TNTNN
```

示例说明： 在第一个测试用例中，只需正确评估至少一个预测即可：4-1 = 由于所有区间的交集为空，因此拒绝所有预测至少能保证一次评估正确。

在第二个测试案例中，我们必须至少正确评估以下预测：5-1 = 2。例如，若AGI在*时间点*$x = 2$出现，则预测1和2成立，因此我们正确评估了预测1、4、5。*若*$x = 5.7$，则所有预测均成立，我们仅正确评估了预测1、3，但这仍满足要求。可证明无论x取值如何，至少有2个预测会被正确评估

。

问题B 柯拉兹求

和

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，0.5秒。 *2025年6月21日*

我们定义自然数上的函数 f：

$$f(x)=\begin{cases} x/2, & \text{若 } x \text{ 为偶数,} \\ 3x+1, & \text{若 } x \text{ 为奇数。} \end{cases}$$

令 $g(x)$ 为函数 f 自身 k 次迭代的结果：

$$g(x)=\underbrace{f(f(\ldots f(x)\ldots))}_{k\text{ times}}.$$

你的任务是，*给定*N和k，计算和

$$S=\sum_{x=1}^{N} g(x) \pmod{10^9+7}。$$

输入

一行包含两个整数 N 和 k（$1\le N\le 10^{12}$，$0\le k\le 32$）。

输出

输出单个数值——*求和*结果S除以10^9的余数。+ 7.

示例

对于输入：
```
10 2
```
正确输出应为：
```
73
```

而对于输入：
```
999888777666 1
```
正确输出应为：
```
990122835
```

示例说明：

在第一个示例中，函数 $g(i)$ 对连续 i 的值为：2, 4, 5, 1, 8, 10, 11, 2, 14, 16。其之和为73。

在第二个示例中，S= 874805371740356549439861。

问题C

DNA子序列

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，3秒。 *2025年6月21日*

本任务研究DNA分子中的核苷酸序列，即由字符‘A’、‘C’、‘G’和‘T’组成的字符串。对于任意自然*数k*，存在4^k种不同的k字母核苷酸序列。对于固定的自然数k，若给定核苷酸序列s中所有k字母序列均为s的子序列（不要求连续），则称s是*k富集序列*。

给定一个由n个字母组成的核苷酸序列*s*。对于[1，*n*]范围内的每个自然*数k*，输出使s成为k富集序列所需修改的最小字符数。注意：每个*k*的计算结果均独立计算。

输入

输入的第一行包含一个*整数n*（1≤ *n*≤ 200 000），表示字符*串s*的长度。

输入的第二行包含一个由n个字母组成的核苷酸序列*s*，仅由字符‘A’、‘C’、‘G’和‘T’。

输出

输出应*包含n*个整数；第k个整数应表示为*使s*成为k富集核苷酸序列所需修改的最小字符数。若*给定k值下*无法按描述方式修改*s中的字符*，则第k个数应改为−1。

示例

输入数据:	正确结果为:
8 AAGTAGAA	1 3 -1 -1 -1 -1 -1 -1

解释： 当*k*=1时，我们可通过一次修饰将*s*变为AAGTCGAA。此时生成的核苷酸序列包含所有单字母子序列（即四个字母各出现至少一次），因此属于1-富序列。

当k= 2时，可通过三次修改将s变为富含2的核苷酸序列CAGTTGAC。需注意*无法*将其改为CCGTTGAA，因该序列不包含富含2的子序列AC。

当*k*> 2时，无法使*序列s*成为k富含*序列*。

问题 D 蕨类市场

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，3秒。 *2025年6月21日*

未来n天内，拜城将开设蕨类植物市场。第i天的蕨类价格预定为a_i拜塔拉。每日你可选择按给定价格买卖蕨类，但受物流限制，每日最多只能交易一株。你也可选择在任意一天不进行交易。显然，若你当时没有蕨类植物则无法出售。但你可以无限量囤积蕨类植物，等待最佳时机出售。你拥有充足的储蓄，蕨类投资资金不会短缺。在第1天开始时你没有蕨类植物，且希望在第n天市场关闭时清空所有蕨类库存。

定义函数 $f(a_1, a_2, ..., a_n)$ 为蕨类植物交易的最大可能利润（单位：比塔拉），前提是提前对购买与销售进行最优规划。然后，对于任意自然数 n，令 $g(n)$ 为所有 $n!$ 种整数 1 到 n 的排列 p 所对应的 $f(p)$ 之和。

给定两个自然数 k 和 m，其中 m 是质数。输出每个值 $g(1)$、$g(2)$、...、$g(k)$ 按模 m 除法所得的余数。

输入

输入的第一行也是唯一一行包含两个整数 k 和 m（$1 \le k \le 7\,000$; $10^8 + 7 \le m \le 10^9 + 7$; m 是素数），正如问题陈述中所描述的。

输出

输出应包含k行。第i行应包含值$g(i)$模m的余数。

示例

对于输入数据:	正确结果为:
4 1000000007	0 1 8 64

解释： 若$n = 1$，则市场仅持续一天。由于初始时没有蕨类植物，且希望结束时也没有蕨类植物，因此无法进行任何交易。

若 $n = 2$，则存在两种可能的价格序列：[1, 2] 和 [2, 1]。第一种情况：先以1比特拉购买蕨类植物，再以2比特拉售出，获利1比特拉。故$f(1,2)=1$。第二种情况：无法获利，即$f(2,1) = 0$。

当$n = 3$时，存在六种可能的价格序列：

- $f(1, 2, 3) = 2$,
- $f(1, 3, 2) = 2$,
- $f(2, 1, 3) = 2$,
- $f(2, 3, 1) = 1$,
- $f(3, 1, 2) = 1$,
- $f(3, 2, 1) = 0$.

在前三个案例中，你可以用一比塔拉买下蕨类植物，再以三比塔拉的价格售出。第四个案例里，你可以用两比塔拉购入蕨类植物，转手以三比塔拉售出。第五个案例中，你可用一比塔拉买入蕨类植物，再以两比塔拉的价格售出。而在最后一个案例里，你无法获得任何利润。

请注意，你可以存储多于一株蕨类植物。例如，$f(2, 1, 4, 3) = 4$，因为你可以在第一天和第二天购买蕨类植物，然后在第三天和第四天出售它们。

问题E 一位

元

第三届万国杯第40关：Potyczki。限制：1024 MB内存，2秒时间。 *2025年6月21日*

你被给定一个由非负整数组成的序列 a_1 ，a_2 ，．．．．，a_n。若序列中任意两个相邻数的二进制表示相差不超过一位*，则该序列称为*非凡序列*。你可以选择序列中的任意元素，将其替换为任意非负整数。要使给定序列成为非凡序列，至少需要修改多少个元素?

输入

输入的第一行包含一个整数 n（$1\leq n\leq 300\,000$)，表示序列的长度。

≤≤输入的第二*行包含*n个*整数*a_1、a_2、．．．、a_n（$0 \leq a_i \leq 300\,000$，－1）。

输出

*输出*应包含一个整数__序列 a_1，a_2，．．．，a_n中元素的最小数量。
需要进行修改才能使这个序列变得引人注目。

示例

对于输入数据:

```
5
4 0 3 3 10
```

正确结果应为:

```
2
```

解释： 例如，我们可以交换第三个和第四个元素，得到显著序列[4，0，2，10，10]。可以证明，仅改变一个数字无法使序列成为显著序列。

*比较不同长度的二进制数时，需在较短数的前端补零以匹配长度。例如比较 $3_{(10)}= 11_{(2)}$ 与 $16_{(10)}= 10000_{(2)}$ 时，需在前数补零形成二进制序列 00011。二进制序列10000与00011在三位上不同，因此数字3与16相差三位。

问题F 谜题IV

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，2秒。 *2025年6月21日*

你得到一个由n个整数*组成的序列*p_1，p_2，. . . .，p_n，初始*状态为1至n的*排列。你的任务是将其排序为升序。为此，你可以执行两种操作：将元素值加到相邻元素上，或将元素值减去相邻元素的值。

操作过程中，任何元素的值在任何时刻都不得超出[1，n]范围。操作次数不得超过2 500 000次。

输入

输入的第一行包含一个整数 n ($2 \leq n \leq 30\,000$)，表示输入序列的长度。

输入的第二行包含 n 个互不相同的整数 p_1，p_2，. . . .，p_n ($1 \leq p_i \leq n$) —— 待排序的序列。

输出

输出第一行应包含一个整数 r ($0 \leq r \leq 2\,500\,000$)，表示你希望执行的操作次数。

接下来的 r 行应包含连续操作的描述，格式如下：

- a+ b: 将第 b 个元素的值加到第 a 个元素上。
- a - b：从第 a 个元素中减去第 b 个元素的值。

在上述操作中，$|a-b|=1$ 必须始终成立。

请注意，您无需将操作次数最小化。可以证明，通过最多执行2,500,000次操作，总能完成排列排序。

通过最多2 500 000次操作完成排序。

示例

对于输入数据：

```
3
1 3 2
```

正确结果为：

```
3
2 - 3
3+ 2
2+ 1
```

解释： 序列在第一次操作后变为[1，1，2]，第二次操作后变为[1，1，3]，

经过第三次操作后，结果为[1，2，3]。

问题G 家具

第三届万国杯，第40阶段：Potyczki。限制：1024 MB，0.5秒。 *2025年6月21日*

巴伊塔扎尔为公寓订购了N件新家具，计划将其全部摆放在尺寸为$A \times B$的矩形客厅中。每件家具本身也是矩形，第i件家具尺寸为$c_i \times d_i$。

每件家具必须紧贴长度为A的墙壁放置，其长度为c_i的边与该墙壁齐平。

墙壁。当然，两件家具不能重叠；它们最多只能沿侧边或在角落处相触。

请编写程序帮助巴伊塔扎尔判断是否能在客厅内摆放全部N件家具。

你将处理T个独立测试用例。

输入

第一行包含一个整数 T（$1 \le T \le 30$），表示测试用例数量。

每个测试用例的第一行包含三个整数 A、B、N（$1 \le A, B \le 10^6$, $1 \le N \le 1000$），分别表示巴伊塔扎尔客厅的尺寸和订购的家具数量。≤≤接下来每行包含两个整数 c_i, d_i（$1 \le c_i \le 10^6, \le A$, $1 \le d_i \le 10^6, \le B$），表示第 i 件家具的尺寸。

所有测试用例中N的总和不超过1000。

输出

输出 T 行。在第 i 行输出单词 TAK（波兰语中的“是”），表示第 i 个测试用例中所有家具均可放置；否则输出 NIE（波兰语中的“否”）。

示例

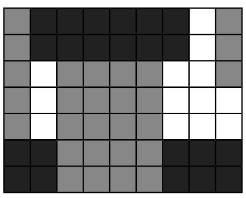
输入数据:

```
4
9 7 6
2 2
3 2
4 5
1 5
1 3
6 2
3 3 3
3 1
1 1
3 1
3 3 3
1 3
1 3
1 3
3 3 2
2 2
2 2
```

正确结果为:

```
TAK
NIE
TAK
NIE
```

示例说明：第一个测试用例中家具的有效摆放方式如下所示：



问题H 合并排

序

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，3秒。 *2025年6月21日*

*归并*排序是最著名的排序算法之一。本题采用以下伪代码描述的实现方案：

算法： 归并排序算法实现。

1: **函数** MERGESORT(p, n) ◁返回列表 p 的排序版本= [p[0], p[1], …, p[n− 1]]。
2: **if** $n = 1$ **then**
3: **返回** p
4: left← MERGESORT([p[0], . . . , p[⌈n/2⌉− 1]], ⌈n/2⌉)
5: 右← MERGESORT([p[⌈n/2⌉], . . . , p[n− 1]], ⌊n/2⌋)
6: (i, j)← (0, 0)
7: result← [] ◁[]表示空列表。
8: **while** i <| left| **and** j <| right| **do** ◁|左|，| 右| 是列表left、right的长度。
9: **如果** left[i]< right[j] **则**
10: 将左列表的值 left[i] 追加到结果列表末尾。
11: i← i+ 1
12: **else**
13: 将值 right[j] 追加到列表 result 的末尾。
14: j← j+ 1
15: 将值 left[i], . . . , left[| left| − 1] 追加到列表 result 的末尾。
16: 将右侧数组右[j], . . . , 右[| 右| − 1]的值追加到结果列表末尾。
17: **返回** result

+已知数字 n、a 和 b。你的任务是统计从 1 到 n 的数字排列中，使用 MERGESORT 程序排序时，在第9 行对 a 和 b 至少进行一次比较的排列数量。输出此类排列数量除以 1 000 000 007（10^9+ 7）的余数。

请注意，当数字 a 和 b 在合并排序过程的任何递归调用中被比较时，都应计入排列次数，而不仅限于最浅层的调用。另请注意，我们允许数字 a 和 b 以任意顺序进行比较。换言之，仅需满足条件 $a < b$ 或 $b < a$ 即可计入排列次数。

此外，您需要解决多个测试用例。

输入

输入的第一行包含一个整数 t（$1 \leq t \leq 10\,000$)，表示测试用例的数量。

接下来的 t 行描述连续的测试用例。第 i 行包含三个整数 n、a 和 b（$2 \leq n \leq 1\,000\,000$; $1 \leq a, b \leq n$; $a \neq b$），具体描述如任务所述。

输出

输出应包含 t 行。第 i 行应包含一个整数——第 i 个测试用例中所需排列数的 $10^9 + 7$ 除法的余数。

示例

对于输入数据:	正确结果为:
3 4 2 3 5 4 1 3 1 3	24 52 4

解释： 在第一个测试用例中，在对所有长度为4的排列进行排序时，会比较数字2和3。

在第三个测试用例中，我们将比较数字1和3在排列[1，2，3]、[1，3，2]、[2，1，3]和[3，1，2]的排序过程中。

问题 I 圣诞老人

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，1秒。 *2025年6月21日*

圣诞老人肩负重任：不仅要准时将礼物送达所有孩子手中，还必须确保每个孩子收到对应的礼物。乖孩子会得到玩具、算法挑战赛决赛门票或自行车，而淘气孩子则会收到一块煤或Windows Vista安装盘。

为按时完成任务，今夜圣诞老人必须向恰好K名儿童派发礼物，但精灵们忘记标注收件人姓名！他必须自行破解难题——*从N份礼物中挑选K份*派发，并分配给M*名*等待儿童中的K人。

圣诞老人知晓每份礼物的质量 j_i（可能为正值或非正值），以及每个孩子的乖巧度 g_i（同样可能为正值或非正值）。

圣诞老人希望最大化礼物分配的公平性：若*礼物*i分配给孩子k，则总公平度*增加*$j_i \cdot g_{(k)}$。
你能帮他确定分配恰好K份礼物时可能达到的最大公平度吗?

输入

第一行包含*三个整数*K、N*和*M（$1 \leq K, N, M \leq 200\,000$, $K \leq \min(N, M)$），分别表示圣诞老人今日要送出的礼物数量、仓库中的礼物数量以及等待礼物的儿童数量。
等待礼物的孩子数量。

第二行包含N个*整数*$j_1, ..., j_N$($-10^6 \leq j_i \leq 10^6$)，表示礼物的质量。第三行包含 M 个整数 $g_{(1)}, ..., g_{(M)}$ ($-10^{(6)} \leq g_{(i)} \leq 10^{(6)}$)，表示孩子的乖巧程度。

输出

输出一个整数——分配 K 件礼物时可能达到的最大公平度。

示例

对于以下输入:

```
3 3 5
2 -2 4
-10 0 -2 3 2
```

正确的输出是:

```
36
```

对于输入:

```
1 4 4
0 0 0 0
2 4 -2 0
```

正确输出应为:

```
0
```

对于输入:

```
1 2 2
1 2
-1 -2
```

正确输出应为:

```
-1
```

示例说明:

在第一个示例中，所有礼物应按以下方式分配:

- 质量为2的礼物应赠予善良值为2的孩子。
- 质量为- 的礼物2应赠予友善度为- 的10岁儿童。
- 品质为4的礼物应赠予善良值为3的孩子。总公平度为$2 \cdot 2 + (-2) \cdot (-10) + 4 \cdot 3 = 36$。

在第二个例子中，仓库里的所有礼物都完全平庸，因此无论选择哪一个
礼物或送给谁都无所谓——公平度永远是0。

在第三个例子中，所有孩子都调皮捣蛋，所有礼物都极具吸引力。我们能做的最好安排是：把最不吸引人的礼物送给最不调皮的孩子。

问题J 住宅区

第三届环球杯第四十关：波蒂茨基。限制：1024 MB，2秒。 *2025年6月21日*

字节宝公司计划建造一个由公寓楼组成的住宅区。该公司已购得一块矩形土地，将其划*分为n*行m列，从而将土地分割成n×m个矩形单元。为简化起见，用坐标(i， j)表示位于第i行第j列的单元。
该图的第j行和第j列。他们需要选择某些单元格（可能为零或全部）并在每个单元格上建造公寓楼。
你的任务是决定在哪些单元上建造公寓楼，并确定每栋楼的高度（以层数计）。

设计过程并非简单！根据官方分区条例，坐标(i, j)处的建筑高度不得超过$h_{(i,j)}$。特别地，*若*$h_{i,j} = 0$，则该单元格不可建造。此外，ByteBud已为建筑群间区域制定了初步规划。具体而言，对于地块上每个四单元交汇点，他们已规划好位于这些单元上的建筑群高度之和。你将获得每对索引对应的数值 $s_{(i,j)}$。
$i \in \{1, ..., n-1\}$, $j \in \{1, ..., m-1\}$; 该数值规定坐标为 (i, j)、$(i+1, j)$、$(i, j+1)$、$(i+1, j+1)$ 的单元格所含区块总数必须精确为 $s_{i,(j)}$ 层。

要求很多，ByteBud今天就需要住宅区设计*方案*……因此，请检查是否存在满足所有要求的住宅区设计方案——若存在，请提出任何有效的方案！

输入

输入的第一行包含两个整*数n和m*（$2 \le n, m \le 300$），分别表示ByteBud将地块划分的行数和列数。

然后，*以下*n行中的第i行包含m个*整数*$h_{i,1}, ..., h_{i,m}$($0 \le h_{i,j} \le 10^{10}$)；第j个数
$h_{i,j}$ 表示坐标为(i, j)的单元格上公寓楼的最大高度（以层数计）。
*接下来的*n行（-1）描述了建筑群之间的初步空间规划。其中第i行*包含*m个整数（-1）：$s_{i,1}, ..., s_{i,m-1}$($0 \le s_{i,j} \le 10^{10}$); 第 j 个数 $s_{i,j}$ 规定坐标为 (i, j)、$(i+1, j)$、$(i, j+1)$、$(i+1, j+1)$。

输出

若存在满足所有要求的住宅区设计方案，则在输出第一行打印TAK。随后在接下来的*n*行中给出该设计方案的示例。第i行应包含m个非负整数$x_{(i,)(1)}, ..., x_{(i,m)}$，表示坐标为(i, 1), ..., (i, m)的单元格上公寓楼的高度（以层数计）。

若不存在符合要求的方案，则在输出唯一一行打印NIE。打印TAK或NIE时，每个字符均可使用小写或大写形式。

示例

对于输入数据:
```
3 4
5 7 9 4
3 4 9 6
9 0 8 5
11 26 20
13 16 18
```
一个正确的示例结果是:
```
TAK
0 6 9 3
1 4 7 1
8 0 5 5
```
然而，对于输入数据:
```
2 2
0 0
0 0
1000
```
正确结果应为:
```
NIE
```

问题K项目

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，2秒。 *2025年6月21日*

您是某软件公司的经理。您手下n名程序员今日需完成m个项目。*项目*i仅能由*程序员*a_i和b_i共同完成。*程序员*a_i*将在*t_i时间内完成该项目。字节秒，*且程序员*b_i（*若*a_i/= b_i）在此项目上的工作效率可能降低至原来的二分之一。两位程序员可分工协作：程序员$a_{(i)}$*可*选择实数$x_{(i)}$ $(0 \leq x_{(i)} \leq \leq x_{(i)} \leq t_{(i)})$，并为项目i投入$x_{(i)}$字节秒。此时程序员$b_{(i)}$需为该项目投入$2\cdot(t(i)-x(i))$字节秒。可能出现$a_{(i)} = b_{(i)}$的情况，此时程序员$a_{(i)}$必须独自完成整个项目，耗费$t_{(i)}$字节秒。

程序员 a_i和 b_i 可独立处理项目 i。例如，他们可同时工作，或在完全不同的时间段分配工作时间。

每位程序员每次只能处理一个项目。程序员的总工作时间为所有项目耗时之和。我们假设项目切换耗时可忽略不计。

今天所有程序员将同时开始工作。其中一些人会告诉你他们很着急，希望尽早下班。作为优秀的管理者，你希望满足这些期望。然而，你尚未确切知晓哪些程序员急于完成工作。因此，你需要考虑q个独立场景。每个场景由急于完成工作的程序员组成的非空子集描述。对于每个场景，确定最小的实数T，使得你可以安排程序员的工作，最终完成所有项目，且每位急于完成工作的程序员工作时间不超过T字节秒。

可以证明结果将是有理数。因此请将所有答案表示为不可约分数。

输入

输入的第一行包含*三个整数*n、m*和*q（$1 \leq n \leq 13$; $1 \leq m \leq 200$; $1 \leq q \leq 10\,000$），分别表示程序员数量、项目数量和待考虑场景数量。
分别表示程序员数量、项目数量和需考虑的场景数量。

接下来的m行，每行包含三个*整数*a_i、b_i*和*t_i（$1 \leq a_i$、$b_i \leq n$; $1 \leq t_i \leq 1\,000\,000$），分别表示负责某个项目的程序员*以及程序员*$a_i$完成该项目所需的时间。

接下来的 q 行描述场景；第 i 个场景占一行，包含一个长度为 n 的二进制字符串 s_i；若第 j 个程序员赶时间，则第 j 个字符为'1'，否则为'0'。每个二进制字符串 s_i 至少包含一个字符'1'。

输出

输出应包含q行；第i行应包含一个*有理数*T，其形式为不可约*分数*x/y（其中$GCD(x, y) = 1$*且*$y > 0$）——这是第i种场景下匆忙程序员的工作时间所能达到的最小极限。

示例

对于输入数据:	正确结果为:
5 7 7	0/1
2 1 2	1/1
2 2 1	4/1
3 2 3	7月18日
3 4 5	3月28日
4 3 2	19/4
1 5 7	19/4
1 5 7	
10000	
01000	
00110	
11100	
11111	
01111	
01111	

说明:

在第一种情境中，程序员1可以立即离职，因为其他程序员能够独立处理项目。

在第二种情景中，程序员2必须完成第二个项目，这将耗费他1字节秒。

在第三种情景中，程序员4将花费2字节秒处理项目五，并花费2字节秒处理项目四。程序员3则应花费4字节秒处理项目四。

在第四种情境中，程序员1和程序员3将各自花费$\frac{18}{7}$字节秒处理项目一和项目三，程序员2分别需要花费$\frac{5}{7}$、1和$\frac{6}{7}$字节秒完成前三个项目，项目，工作日结束时间为$\frac{5+7+6}{7} = \frac{18}{7}$。

第五种情景中，程序员1和5各自需至少耗费$\frac{7}{3}$字节秒完成最后两个项目项目上花费至少$\frac{7}{3}$字节秒。存在一种策略，每位程序员最多在$\frac{28}{3}$字节秒后完成工作。

问题L 有向汉诺塔

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，2秒。 *2025年6月21日*

定向河内塔游戏的棋盘由*N*个编号*为1至N的*柱钉构成。部分柱钉间存在定向边，始终从编号较小的柱钉指向编号较大的柱钉。初始状态下，编号1的柱钉上放置着*K*个圆盘，尺寸从*1至K*，按从大到小顺序（自下而上）排列。游戏目标是将所有圆盘移至*编号N的*柱钉上。

在单次移动中，你可以将插在某根*柱子A*顶端的圆盘移至任意*另一根柱子B的*顶端，前提是：*柱子B*要么是空的，要么*其顶端*圆盘的直径大于被移动的圆盘；且从*柱子A*到*柱子B*的路径上（不要求仅有一条路径）的其他柱子上的圆盘大小无关紧要。

你的任务是：给定一个有向的河内棋盘，确定*最大的K*值，使得存在一种方法能实现游戏目标（即把*所有K*个圆盘移到*第N根*柱子上）。可以证明，圆盘的最大数量总是有限的。

输入

输入的第一行包含两个整数：*N*（$2\leq N\leq 500$）*和M*（$0\leq M\leq N(N-1)/2$），分别表示柱数和边数。

*接下来的*M行每行描述一条边，包含两个*整数*a_i*和*b_i（$1\leq a_i< b_i\leq N$），*表示钉子*a_i*和*b_i由一条有向边连接。每对钉子在输入中最多出现一次。

输入中出现一次。

输出

输出单个整数——在输入描述的木桩上进行有向汉诺塔游戏时，能够*将K*个圆盘从木桩1移动到木*桩N的*最大可能*值K*。

示例

对于输入：
```
5 4
1 2
2 3
3 4
4 5
```
正确输出应为：
```
5
```

对于输入：
```
6 8
1 2
2 3
3 6
1 4
4 5
5 6
2 5
1 6
```
正确输出应为：
```
5
```

对于输入：
```
2 0
```
正确输出为：
```
0
```

示例说明：

在第一个示例中，将5个圆盘从第1根柱子转移到第5根柱子的操作序列如下所示：

柱1	54321	5432	543	543	54	5						
挂钩2		1	1		3	3	3	3	3			
挂钩3			2	21	21	21	21	21	2	2		
挂钩4						4	4		1	1	1	
挂钩5							5	54	54	543	5432	54321

在第三个例子中，没有边，因此无法移动任何圆盘。

问题M俱乐部

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，1秒。 *2025年6月21日*

俱乐部是一款双人合作游戏，玩家需协作达成*最优合约*。双方各自持有部分为梅花花色的牌组，各自知晓自身梅花数量但未知对方持有量。最优合约取决于双方手牌中梅花总数：若*总数小于*X，则最优合约*为N*；否则为N+ 1.

合约通过竞标达成。首位玩家率先提出合约——一个正整数。随后玩家轮流行动。轮到自己时，玩家可选择接受上次提出的合约，或提出更高的新合约（同样为正整数）。当某位玩家接受合约时竞标结束；若该合约为最优方案，则竞标成功。

阿尔戈西亚和巴杰克正在玩梅花牌。*他们都知道X*、N，并且知道阿尔戈西亚手中的梅花数量*是*a_1、a_2、……、$a_{(k)}$*中的*一个值，而巴杰克手中的梅花数量*是*$b_{(1)}$、b_2、……、$b_{(k)}$*中的*一个*值*。游戏开始前，两人各自得知了自己手中黑桃的精确数量。

他们致力于寻找*最小值N*，使得存在一种能保证成功的竞标策略。Algosia始终是第一玩家。可以证明，这样的最小*值N*确实存在。

输入

第一行包含三个整数 X、k 和 l（$1 \leq X \leq 10^9$；$1 \leq k, l \leq 1000$），分别表示双方手中*实现N*+ 1*最*优契约所需的俱乐部总数，以及Algosia和Bajtek手中可能的俱乐部数量之和。
Algosia与Bajtek手中黑桃可能的计数数量。

第二行包含k 个*整数*$a_1, ..., a_k$($1 \leq a_1 < a_2 < ... < a_k \leq 10^9$), 表示阿尔戈西亚手中可能持有的梅花牌数量。第三行包含 l 个整数 $b_{(1)}, . . . , b_{(l)}$ ($1 \leq b_{(1)} < b_{(2)} < . . . < b_{(l)} \leq 10^{(9)}$)，表示巴杰克手中可能持有的梅花数量。

输出

输出一个整数——使得存在策略能保证Algosia和Bajtek达到最优契约的最小*N值*。

示例

对于以下输入：

```
13 5 3
1 5 7 10 12
2 6 9
```

正确输出应为：

```
3
```

而对于输入：

```
2 1 1
1
1
```

正确输出应为：

```
0
```

示例说明：

在第一个示例中，当$N = 3$时，可采用以下策略：

- 若Algosia持有5、7或10张梅花，则建议1级契约。
  - Bajtek若持有6张梅花则应答2。
    - *阿尔戈西亚若持有5张梅花则回应3。巴杰克接受（双方共计11张梅花）。
    - *若阿尔戈西亚持有7或10张梅花牌，则应答4。巴杰克接受（此时双方共计持有13或16张梅花牌）。
  - 巴杰克若持有2张梅花则应答3。阿尔戈西亚接受（双方共计7、9或12张梅花）。
  - 巴杰克若持有9张梅花则应答4。阿尔戈西亚接受（双方共计14、16或19张梅花）。
- 阿尔戈西亚若持有1或12个俱乐部则出价2。无论巴杰克持有多少俱乐部，他都以3回应。
  - 若阿尔戈西亚持有1根棍棒，则接受（此时总棍棒数为3、7或10根）。
  - 若阿尔戈西亚持有12张梅花，则应叫4。巴杰特应叫（此时双方共计持有14、18或21张梅花）。可证明当$N = 2$时不存在合理策略。

在第二个例子中，当N= 0时，阿尔戈西亚叫1= ，N+ 1，巴杰特接受该契约。

问题N频率函数

第三届环球杯，第40阶段：Potyczki。限制：1024 MB，2秒。 2025年6月21日

定义函数 $f(a)$，其参数为 n 个整数序列 $a_1, a_2, ..., a_n$，取值范围为$[0, n]$，并返回序列 $b_1, b_2, ..., b_n$，使得 b_i 表示数字 i 在序列 $a_1, a_2, ..., a_n$ 中出现的次数。

此外，我们定义其k阶复合运算：

$$f^k(a) = \begin{cases} a & \text{当 } k = 0 \text{ 时} \\ f(f^{k-1}(a)) & \text{当 } k > 0 \text{ 时} \end{cases}$$

你被给定一个序列 $a_1, a_2, ..., a_n$。你的任务是处理两种类型的查询：

- 1 v x – 将数列元素 a_v 修改为 x。
- 2 k v – 输出序列第 v 个元素 $f^k(a)$。

输入

输入的第一行包含两个整数 n 和 q（$1 \le n \le 300\,000$; $1 \le q \le 500\,000$），分别表示输入序列的长度和查询的数量。

输入的第二行包含一个由n个整数组成的序列：$a_1, a_2, ..., a_n$（$0 \le a_i \le n$）。

≤≤接下来的 q 行按题目要求格式描述查询：满足 $1 \le v \le n$, $0 \le x \le n$ 且 $0 \le\le k \le 300\,000$。

可以保证至少存在一个第二类查询。

输出

输出应包含与第二类查询数量相同的行数。第 i 行应包含一个整数——即第 i 个第二类查询的答案。

示例

对于输入数据：

```
6 6
2 1 2 3 0 3
2 3 2
1 5 2
2 0 2
1 2 3
2 0 2
2 2 3
```

正确结果为：

```
1
1
3
2
```

说明： 让我们分析最后一个查询。我们有：

- $f^0(a) = [2, 3, 2, 3, 2, 3]$,
- $f^1(a) = [0, 3, 3, 0, 0, 0]$,
- $f^2(a) = [0, 0, 2, 0, 0, 0]$.

此查询的答案为 2。