第三届环球杯

第 39 赛段东京

2025 年 6 月 7-8 日

这套问题应包含 17 个问题（A 至 L），共 34 页。

问题 A.数组相似性

时间限制 2 秒 内存限制 1024 兆字节

设 $a=(a_1, a_2, \ldots, a_n)$ 和 $b=(b_1, b_{(2)}, \ldots, b_n)$ 是两个长度相等的序列。我们说 a 和 b 是当且仅当对于每个 $i=1, 2, ., n$,

$a_i = \max(a_1, a_2, \ldots, a_i)$ 当 $b_i = \max(b_1, b_2, \ldots, b_{i)}$ *恰好成立。*

给你一个序列 $(A_1, A_2, \ldots, A_{N)}$。回答 Q 个查询。在*第* i 个查询中，您会得到整数 $L_{(i,)\,(1)}, R_{(i,)\,(1)}, L(_{i,)\,(2)}, R(_{i,)\,(2)}$.判断两个子序列

$$(A_{(L)\,(_{i,})\,(1)}, A_{(L)\,(_{i,})\,(1)\,(}+1), ., A_{(R)\,(_{i,})\,(1)}) \text{ 和 } (A_{(L)\,(_{i,})\,(2)}, A_{(L)\,(_{i,})\,(2)\,(}+1), ., A_{(R)\,(_{i,})\,(2)})$$

相似。

输入

输入格式如下

```
N Q
A1A2... ...  A(n)

L1,1    R1,1   L1,2   R1,2
L2,1    R2,1   L2,2   R2,2

.
LQ,1
        RQ,1   LQ,2   RQ,2
```

- 所有输入值均为整数。
- $1 \le n \le 2 \times 10^5$.
- $1 \le q \le 2 \times 10^5$.
- $1 \le A_i \le 10^9$.
- $1 \le L_{i,1} \le R_{(i,)\,(1)} \le N$.
- $1 \le L(_{i,)2} \le R_{(i,)\,(2)} \le N$.
- $Ri_{,1} - {}_{Li,1} = Ri_{,2} - {}_{Li,2}$。

输出

打印 Q 行。在*第* i 行，如果子序列

$$(A_{(L)\,(_{i,})\,(1)}, A_{(L)\,(_{i,})\,(1)\,(}+1), \ldots, A_{(R)\,(_{i,})\,(1)}) \text{ 和 } (A_{(L)\,(_{i,})\,(2)}, A_{(L)\,(_{i,})\,(2)\,(}+1), ., A_{(R)\,(_{i,})\,(2)})$$

相似；否则，打印 "否"。

示例

标准输入	标准输出
10 6	是
3 1 4 1 5 9 2 6 5 3	编号
1 3 3 5	有
1 5 6 10	是
1 1 9 9	是
1 9 1 9	有

1 3 6 8 5 8 7 10	

备注

在第一个查询中，(3, 1, 4) 和 (4, 1, 5) 相似，因此输出为 "是"。

在第二个查询中，（3, 1, 4, 1, 5）和（9, 2, 6, 5, 3）不相似，因此输出为 "否"。在第三个查询中，注意 $L_{(i,)\,(1)}=R_{(i,)\,(1)}$和 $L_{(i,)\,(2)}=R_{(i,)\,(2)}$ 是可能的。

在第四个查询中，注意有可能出现 $L_{(i,)\,(1)}=L_{(i,)\,(2)}$和 $R_{(i,)\,(1)}=R_{(i,)\,(2)}$ 的情况。

问题 B. 括弧字符频率

时间限制 2 秒 内存限制 1024 兆字节

当且仅当一个字符串 S 只由 (和) 字符组成，且满足以下任一条件时，该字符串 S 才被称为**正确的括号序列**。

- S 是空字符串。
- S 由 (, A,) 按此顺序连接而成，其中 A 是正确的括号序列。
- S 是由 A 和 B 按此顺序连接而成，其中 A 和 B 都是正确的括号序列，且都不是空字符串。

给你整数 N、K 和长度为 $2K$ 的整数序列 $A=$ （$A_1, A_2, ---, A_{(2)(K)}$）。

请判断是否存在满足以下条件的 N 个正确括号序列的元组。

- N 个正确的括号序列的长度都是 2K。
- 对于 $i=1, 2, ---, 2K$，在 N 个正确的括号序列中，正好有 A_i 的第 i 个字符是（。

给你 T 个测试用例。请分别回答每个测试用例。

输入

输入的格式如下：

```
T
案例1案
例2
.
案例T
```

每个测试用例的格式如下：

```
N K
A(1) A(2) -- A(2)(K)
```

- 所有输入值均为整数。
- $1 \leq T \leq 10^5$.
- $1 \leq N \leq 10^{12}$.
- $1 \leq k \leq 2 \times 10^5$.
- $0 \leq A_i \leq N$.
- 在单个输入的所有测试用例中，K 的总和最多为 5×10^5.

输出

打印 T 行。第 i 行应包含第 i 个测试用例的答案。具体来说，如果有 N 个正确的括号序列组成的元组满足条件，则打印 "是"；否则，打印 "否"。

示例

标准输入	标准输出
2 3 3 3 2 2 0 2 0 3 3 3 0 2 3 1 0	是 是

注释

在第一个测试用例中，3 个正确的括号序列（)()()、 （(())）、 （()()） 满足条件。在第二个测试案例中，没有满足条件的 3 个正确括号序列的元组。

问题 C. 纸牌

时间限制 2 秒 内存限制 1024 兆字节

有 10^{100}张牌，编号从 1 到 10^{100}，堆叠在一起，这样第 i 张牌就是最上面的第 i 张牌。有一个空袋子。你要进行下面的操作 M 次：

> 查看最上面的 K 张牌，从中任选一张（可能是零），然后把选中的牌放进袋子里。将未选中的牌按原来的相对顺序放回牌顶。

经过所有 M 次操作后，考虑袋子里可能有的每一组牌。计算这些集合的大小之和，然后输出该和的模数 998244353。

给你 T 个测试用例。请为每个测试用例输出所需的值。

输入

输入的格式如下

```
T
case_1
.
case_T
```

每个测试用例如下

```
K M
```

- 所有输入值均为整数。
- $1 \le T \le 10^5$.
- $1 \le K < 998244353$.
- $1 \le M < 998244353$。

输出

输出 T 行。在第 i 行打印第 i 个测试用例的答案。

示例

标准输入	标准输出
3 2 1 3 2 20250308 410338673	4 81 509595821

注释

在第一个例子中，袋子里可能有的纸牌组是 {}、{1}、{2}、{1, 2}，它们的大小之和是 4。

问题 D. 前缀积的位数

时间限制 2 秒 内存限制 1024
兆字节

这是一个**只有输出的**问题。不提供输入。

输出一个由 10 个正整数组成的序列（$a_1, a_2, ., a_{10}$），并满足以下**所有**条件。所有整数都应以标准十进制表示，不含前导零。

- 每个 $a_1, a_2, ., a_{10}$中都不包含等于 0 的数字
- $a_1, a_2, ., a_{10}$至少有 100 位数字
- $a_1, a_2, \ldots, a_{(10)}$中的数字总数至多为 0。，$a_{(10)}$ 中的数字总数最多为 10^5
- 对于每个 i（1 i≤ ≤ 10），设 $b_i = a_1 \times a_2 \times \cdots \times a_{(i)}$。那么，在每个 b_i中，每对相邻的数字必须是不同的（没有两个相邻的数字是相同的）。

输入

不提供输入。

输出

按以下顺序输出序列 $a_1, a_2, ., a_{10}$的顺序，每行一个数字，使用十进制符号，不含前导零。

注释

将 $(a_1, a_2, \ldots, a_{10})$ 定义为 $(a_1, a_2, ., a_{10})$= (28, 19, 2, 19, 15, 3, 14, 14, 29, 27) 的结果：

- b_1= 28
- b_2= 532
- b_3= 1064
- b_4= 20216
- b_5= 303240
- b_6= 909720
- B_7= 12736080
- B_8= 178305120
- B_9= 5170848480
- B_{10}= 139612908960

该输出满足条件 1、3 和 4。但是，它**不**满足条件 2，因此被判定为不正确。

问题 E. 边缘着色问题

时间限制 3 秒 内存限制 1024 兆字节

给你一个有 N 个顶点和 (N) (() (N)(-)(3))条边的简单无向图。该图由 N 个二进制字符串 $S_1, S_2,$.其中，如果顶点 i 和顶点 j 之间有一条边，S_i的第 j 个字符为 1，否则为 0。值得注意的是，S_i的第 i 个字符总是 0。

图中每个顶点的度数正好是 $N-3$。

现在，您需要为图中的每条边分配一个正整数。如果共享一个共同顶点的任意两条边都被分配了不同的整数，那么这种分配就叫做**边着色**。在任何有效的边**着色**中使用的最小最大整数称为图的**边色度数**。

您的任务是确定图形的边色度数，并找到一个有效的边着色来达到这个数值。

输入

输入的格式如下：

$$
\begin{array}{l}
N \\
S_{(1)} \\
S_{(2)} \\
\vdots \\
S_N
\end{array}
$$

- $4 \leq N \leq 300$.
- $S_1, S_2, \ldots, S_N$是长度为 N 的二进制字符串，只包含 0 和 1。
- 给定的图是一个简单的无向图，其中每个顶点的阶数为 $N-3$。

输出

打印边色度数 C，然后打印 $N \times N$ 网格，其中单元格 (i, j) 包含整数 $c_{(i,j)}$分配给顶点 i 和顶点 j 之间的边：

$$
\begin{array}{l}
C \\
c_{1,1}\ c_{1,2} \cdots c_{1,N} \\
c_{2,1}\ c_{2,2} \cdots c_{2,N} \\
\vdots \\
c_{N,1}\ c_{N,2} \cdots c_{N,N}
\end{array}
$$

如果顶点 i 和顶点 j 之间没有边，$c_{i,j}$输出-1。特别是，$c_{i,i}$应始终为 -1.

如果存在多个有效输出，则其中任何一个都被认为是正确的。

示例

标准输入	标准输出
6 011100 101010 110001 100011 010101 001110	3 -1 2 3 1 -1 -1 2 -1 1 -1 3 -1 3 1 -1 -1 -1 2 1 -1 -1 -1 2 3 -1 3 -1 2 -1 1 -1 -1 2 3 1 -1
5 01001 10100 01010 00101 10010	3 -1 2 -1 -1 1 2 -1 3 -1 -1 -1 3 -1 1 -1 -1 -1 1 -1 3 1 -1 -1 3 -1

注

在第一个例子中，顶点 1 与顶点 2、3 和 4 相连。这些边必须分配给不同的整数，因此边的色度数至少为 3。

在输出示例中，连接顶点 1 *和*顶点 2、3 和 4 的边分别被赋值为整数 2、3 和 1。所有共享一个共同顶点的边都有不同的整数。满足边缘着色条件的所有其他顶点也具有相同的属性，边缘色度数为 3。

问题 F. 傅立叶系数

时间限制 8 秒 内存限制 1024 兆字节

这是一个**互动问题**。您的程序将通过标准输入和输出与法官交互。法官的执行时间可能长达 1.3 秒。

给你一个整数 N。法官秘密选择一个函数

$$f(x) := \sum_{k=0}^{N-1} A_k \cos(kx)、$$

其中每个 $A_0, A_1, .., A_{(N)(-)(1)}$ 都是整数，其中 $0 \le A_k < 998244353$。

您必须确定 $A_0, A_1, \dots, A_{(N)(-)(1)}$的值：您将输出 N 对整数 $(X_1, Y_1), \dots, (x_n, y_n)$.每对整数必须满足

$$0 \le X_i \le Y_i < 998244353、\qquad Y_i \neq 0。$$

然后，法官将给出 N 个整数 $Z_1, \dots, Z_N$，其中

$$Z_i = f\ \arccos(X_i/Y_i) \bmod 998244353。$$

Z_i**的详细定义。**在 X_i, Y_i的约束条件下，值 $f(\arccos(X_i/Y_{(i)}))$ 是有理数。把它写成最小值 $P_{(i)}/Q_i$；可以证明 $Q_i \not\equiv 0 \pmod{998244353}$。那么，$Z_i$被定义为满足以下条件的唯一整数 $0 \le Z_i < 998244353$

$$Z_i Q_i \equiv P_{(i)} \pmod{998244353}。$$

这样的 Z_i总是存在且唯一的。

输入

- 所有输入均为整数。
- $1 \le n \le 5 \times 10^5$。

交互协议

这是一个交互式问题。你的程序将通过标准输入和输出与法官交互。首先，从标准输入中读取整数 N：

```
N
```

然后按以下格式输出 N 个查询对（$X_i, Y_{(i)}$），并满足上述约束条件：

```
X(1) Y(1)
X(2) Y(2)
.
X(N) Y(N)
```

如果输出有效，法官将回复 N 行：

```
Z(1)
Z(2)
.
ZN Z(1) Z(2) .
```

如果输出无效，您将收到

```
-1
```

如果收到"-1"，程序必须立即终止。

最后，收到 Z_i后，依次输出隐藏系数 $A_0, A_1, ., A_{(N)\,(-)\,(1)}$依次输出：

```
A(0)
A(1)

.
AN-1
```

注意

- **每次输出操作后，请打印换行符并刷新标准输出。** 如果没 有 刷新，可能会收到 TLE 判决。
- 如果在任何时候产生了无效输出或程序意外终止，则判定为未定义。
- 请在打印答案（或读取"-1"）后立即终止程序。否则，结果将是未定义。
- 额外的换行符或与指定格式的任何偏差都将被判定为无效。
- 判决是非适应性的：值 $A_0, \ldots ., A_{(N)\,(-)\,(1)}$在开始时是固定的，在交互过程中不会改变。

示例

假设 $N= 2$ 和 $(A_0, A_1)= (3, 2)$。可能的交互作用如下所示。

输入	您的输出	说明
2		您读取 N .
	0 1 1 1	您查询了两个有效数对 (X_i, Y_i)。
3 5		法官返回 $Z_1 = 3$，$Z_2 = 5$。
	3 2	您输出恢复的结果（A_0, A_1)。

问题 G. 守护计划

时间限制 2 秒 内存限制 1024 兆字节

二维坐标平面上站着 N 名保安。第 i 名保安站在点 (x_i, y_i).

您可以执行以下操作任意多次（包括零次）：

> 选择保安目前站立的两个点，然后选择连接这两个点的线段上的任意一点。如果所选点上已无警卫，则在该点上安置一名新警卫。

站在点 (a, b) 的警卫会监视位于 x 坐标小于或等于 a 和 y 坐标小于或等于 b 区域内的所有警卫。

不受任何其他警卫监视的警卫称为**必要警卫**。

确定最终配置中必要警卫的最少数量，以及实现该配置所需的最少操作次数。

输入

输入格式如下

```
N
x(1) y(1)
x(2) y(2)
.
x(N) y(N)
```

- 所有输入值均为整数。
- $1 \le n \le 2 \times 10^5$.
- $0 \le x_{(i)}, y_i \le 10^9$.
- $(x_i, y_i)_{(i \neq j)} \neq (x_j, y_j)(i \neq j)$。

输出

在第一行，打印一个整数--所需的最小守护数。第二行，打印一个整数--实现该配置所需的最少操作数。

示例

标准输入	标准输出
5 1 6 2 4 3 3 4 2 6 1	4 1
3 0 0 1 2 2 1	2 0
7 10 49 9 27 59 8 19 22 0 50 25 23 33 13	4 1

备注

在第一个示例中，选择位于（1,6）和（6,1）两个点之间的点（3,4），并在此处设置一个新的防卫点。操作完成后，所需的守护点将是位于 (1,6)、(3,4)、(4,2) 和 (6,1) 的四个守护点。

问题 H. 隐藏序列旋转

时间限制 2 秒 内存限制 1024 兆字节

这是一个**交互式问题**（您的程序通过标准输入/输出与法官交互的问题）。

给你一个整数 N。法官持有一个隐藏序列 $A=(A_0,...,A_{(N)(-)(1)})$ ，长度为 N，其中每个元素都是介于 1 和 10^5之间的整数。请注意，在整个问题中，索引都是以 0 为基础的。

对于整数 $s=0,...,N-1$ 和 $l=1,...,N$ ，定义序列 $A(s,l)$ 如下：

- 长度为 l 的序列，其第 i 个元素为 $A_{(s+(i)()\bmod)(N)}$for $i=0,...,l-1$。

您最多可以按以下格式向法官提出 **20 个查询**：

- 您将输出一个整数对列表 $((s_0,l_0),...,(s_{(k)(-)(1)},l_{(k)(-)(1)}))$ 满足以下约束条件：
  - $1\leq k\leq N$
  - $0\leq s_i\leq N-1$
  - $1\leq l_i\leq N$
  - $\sum_{i=0}^{(k)(-)(1)} l_{(i)}\leq N$
- 对此 作为回应、 法官 法官 返回 所有 指数 $i=0,...,k-1$ 这样 使得 $A(s_i,l_i)$ 在序列中是词典最小值。换 句 话 说，法官返回集合 $\{i|\ 0\leq i<k,\ A(s_i,l_{(i)})=\min_{0\leq i'<k}A(s_{(i')},l_{(i')})\}$。

利用这些查询，确定所有 s 值$=0,...,N-1$，使得 $A(s,N)$ 在词法上最小。换句话说，确定集合 $\{s|\ 0\leq s<N,\ A(s,N)=\min_{0\leq s'<N}A(s',N)\}$ 。

请注意，法官**不是自适应的**，这意味着序列 A 在每个测试用例的交互之前都是固定的。

输入

输入格式如下

```
N
```

- N 是 $1\leq N\leq 10^5$范围内的整数。

输出

确定答案后，按以下格式输出：

```
N
S(0)
S(1)
.
sn-1
```

这里，n 是整数，每个 s_i都是在 $0\leq s_i<N$ 范围内的一个不同整数，并且必须满足以下条件

$$\{s_0,...,s_{(n)(-)(1)}\}=\{s|\ 0\leq s<N,\ A(s,N)=\min_{0\leq s'<N}A(s',N)\}。$$

交互协议

您可以按以下格式向标准输出端输出查询：

```
K S(0)
L(0)
S(1) L(1)

.
sk-1 lk-1
```

确保您的查询满足上述条件。如果查询有效，法官将作出以下回应：

```
k′
i(0)
i(1)

.
ik′-1
```

这里，$k', i_0, i_1, .., i_{k'-1}$ 是整数，$0\leq i_0< i_1<\dots< i_{k'-1}< k$，且

$$\{i_0, i_1, .., i_{k'-1}\} = \{i\mid 0\leq i< k, A(s_i, l_i)= \min_{0\leq i'<k} A(s_{i'}, l_{i'})\}。$$

如果您的查询无效（例如，违反了限制条件或超过了允许的查询次数），法官将给出以下回复：

```
-1
```

如果收到-1，请立即终止程序。

交互注意事项

- 每次打印后务必刷新输出。否则可能导致 "超时 " 判决。
- 如果您的程序在交互过程中产生无效输出或不正常退出，则法官的行为是未定义的。
- 打印出最终答案或收到-1 后，程序必须立即终止。否则可能导致未定义行为。
- 避免在输出中出现不必要的换行或空格，因为这些可能会被视为格式错误。

注意事项

下面是一个交互示例，$N= 6$，隐藏序列 $A= (1, 2, 3, 1, 2, 4)$：

输入	输出	说明
6		N 已给定。
	? 3 0 1 1 1 3 1	查询序列 (1)、(2) 和 (1)。
2 0 2		索引 0 和 2 的序列是 序数最小。
	? 2 0 3 3 3	查询序列 (1, 2, 3) 和 (1, 2, 4)。
1 0		索引为 0 的序列按词典顺序为 最小。
	! 1 0	只有 s= 0 在词典中最小化了 序列 A(s，N)，因此将其打印输出。

问题 I. 插入 AB 或 BA

时间限制 2 秒 内存限制 1024
兆字节

给你两个字符串 S 和 T，它们都由字符 A 和 B 组成。

您可以按任意顺序执行以下两种类型的操作任意多次（包括零次）：

- 在 S 的任意位置插入字符串 AB。
- 在 S 的任意位置插入字符串 BA。

请注意，也可以在字符串的开头或结尾插入。

确定是否有可能通过这些操作将 S 变换成 T。如果可能，请输出这样做所需的最小总成本。

输入

输入的格式如下

S T
X Y

- X 和 Y 是整数。
- S 和 T 只由字符 A 和 B 组成。
- $1 \leq |S| \leq |T| \leq 8000$.
- $1 \leq X \leq 10^9$.
- $1 \leq Y \leq 10^9$.

输出

如果可以将 S 转化为 T，则单行输出所需的最小总成本。如果不可能，则输出-1。

示例

标准输入	标准输出
AB ABAABB 5 3	8
AAAAAA AAAAAA 2 3	0
AAAAA BBBBBBB 9982 44353	-1
aaabbabbbbbbbabbabba aaababbabbbbbbbbbaaaababababbaabba 1 100000	300007

注释

在第一个例子中，S = AB。您可以通过执行以下操作将 S 转化为 T = ABAABB：

- 在 AB 的第 1 和第 2 个字符之间插入 BA，结果是 ABAB。

- 在 ABAB 的第 3 和第 4 个字符之间插入 AB，结果是 ABAABB。

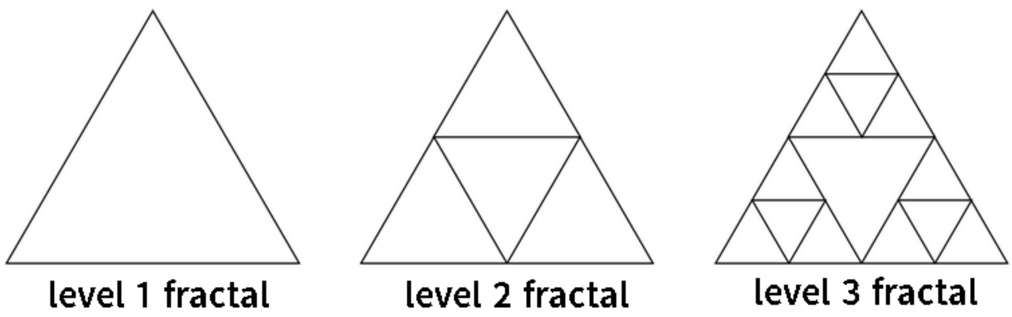
在这种情况下，总成本为 3 + 5= 8，这是实现变换所需的最小总成本。

问题 J. 分形之旅

时间限制 2 秒 内存限制 1024 兆字节

*第 i 层***三角形**是边长为 2^{i-1} 的正三角形。*第 i* **层分形**是根据这些规则生成的图形：

- 第 1 级分形是第 1 级三角形。
- 对于 $i \geq 1$，将三个 i 级分形放置在一个 i 级三角形的两边，使它们完全接触，就形成了一个 i 级 $+ 1$ 分形（见下图）。



给你一个 L 级分形。

爱丽丝首先选取分形中的任意一个三角形。然后，她可以移动到任何与当前三角形共享一条边的未访问三角形。

爱丽丝最多可以移动 K 次。**得分**是爱丽丝移动过的所有三角形（包括起始三角形）的等级总和。

求最大可能得分 modulo 998244353。请注意，您需要计算的是最大可能得分，而不是最大余数。

输入

输入的格式如下

```
L K
```

- 所有输入值均为整数。
- $1 \leq L \leq 10^9$.
- $1 \leq K \leq 10^{18}$.

输出

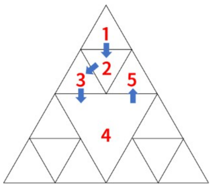
单行打印答案。

示例

标准输入	标准输出
3 4	6
998244353 1000000000000000007	756221200

注释

在第一个测试案例中，爱丽丝可以访问四个第 1 层三角形和一个第 2 层三角形，如下所示：


1
2
3
5
4

问题 K. K-rep 阵列

时间限制 2 秒 内存限制 1024 兆字节

对于正整数 K，如果满足以下条件，由正整数组成的序列 V 称为 *K-rep 序列*:

- 存在一个长度为 K 的由正整数组成的序列 B，使得序列 B' 重复 B 10^{100}次得到的序列中包含连续子序列 V。

给你一个长度为 N 的序列 $A=(A_1, A_2, ..., A_N)$ ，长度为 N，其中每个元素要么是正整数，要么是-1。对于每个 $K=1, 2, ., N$ ，求解以下问题：

- 确定是否存在用一个正整数替换 A 中每个-1 的情况，使得得到的序列是 *K-rep*。

输入

输入的格式如下

```
N
A(1) A(2)... .AN
```

- 所有输入均为整数。
- $1\leq n\leq 2\times 10^5$.
- $1\leq A_i\leq N$ 或 $A_i=-1$ （针对每个 i）。

输出

输出长度为 N 的字符串。如果存在满足 $K=$ i 条件的替换字符，则*第* i 个字符应为 1，否则为 0。

示例

标准输入	标准输出
5 1 2 -1 2 1	01011

备注

在示例中，元素 $A_i=-1$ 的一个可能替换序列是（1，2，3，2，1）。对于 $K=4$，让 $B=(2，3，2，1)$。由于重复 B 得到的序列 B'包含连续子序列 (1, 2, 3, 2, 1)，因此 (1, 2, 3, 2, 1) 是 *K-rep*.

问题 L.LIS 三角形

时间限制 2 秒 内存限制 1024 兆字节

给你三个正整数 N、K 和 L。你的任务是确定是否存在一个长度为 N 的整数序列 P，它满足以下所有条件。如果存在，请输出一个这样的序列。

- P 是序列 $(K, K+1, \ldots, K+N-1)$ 的排列。
- P 的最长递增子序列[1]的长度正好是 L。
- 对于每一个整数 i，使得 $1 \le i \le N-2$，都存在一个非退化三角形[2]，其边长为 P_i, $P_{(i)(+1)}$, 和 $P_{(i)(+2)}$。

给你 T 个测试用例。请分别回答每个测试用例。

输入

输入的格式如下：

```
T
案例1案
例2
.
案例T
```

每个测试用例的格式如下：

```
N K L
```

- 所有输入值均为整数。
- $1 \le T \le 50000$.
- $3 \le N \le 2 \times 10^5$.
- $1 \le K \le 2 \times 10^5$.
- $1 \le L \le N$.
- 在单个输入的所有测试用例中，N 的总和最多为 2×10^5.

输出

如果没有满足所有条件的序列 P，则输出：

```
无
```

否则，输出

[1]序列 P 的**子序列**是指在不改变剩余元素顺序的情况下，从 P 中删除零个或多个元 素 而形成的序列。P 的**最长递增子序列**是 P 的严格递增子序列，具有最大可能的长度。

[2]非退化三角形是指三个顶点不在同一条直线上的三角形。

是 $P_{(1)}$ $_{P(2)}$… .P_N

任何满足所有条件的有效序列 P 都被接受。

示例

标准输入	标准输出
3 6 3 4 5 5 5 7 1 2	是 3 6 4 7 5 8 有 5 6 7 8 9 无

注

在第一个例子中，一个有效序列是 $P = (3,6,4,7,5,8)$。也可能存在其他有效序列。在第二个例子中，唯一有效的序列是 $P = (5$，6

，7，8，9)。

在第三个例子中，没有序列 P 满足所有条件。

问题 M. 最小距离树

时间限制 2 秒 内存限制 1024
兆字节

给你一个连通、无向、加权的简单图 G，它有 N 个顶点，编号从 1 到 N
和 M 条边。第 i 条边连接顶点 u_i和 $v_{(i)}$，权重为 w_i。

判断是否存在一棵有 N 个顶点的加权树 T，其编号也是从 1 到 N，从而对于每一对顶点 u 和 v，G 中 u 和 v 之间的最短路径长度等于 T 中 u 和 v 之间的最短路径长度。

输入

输入的格式如下

```
N M
u(1) v(1) w(1)
u(2) v(2) w(2)

.
u(M) v(M) w(M)
```

- 所有输入值均为整数。
- $2\leq n\leq 5\times 10^5$.
- $N-1\leq M\leq 5\times 10^5$.
- $1\leq u_i, v_i\leq N$.
- $1\leq w_i\leq 10^9$.
- 给定的图是简单且连通的。

输出

如果存在这样一棵树 T，则输出：

```
是
```

否则，输出

```
否
```

示例

标准输入	标准输出
3 3 1 2 3 2 3 4 3 1 100	是
3 3 1 2 3 2 3 4 3 1 2	无

注释

在第一个例子中，有 3 个顶点的树 T 满足条件，其中顶点 1 与顶点 2 相连，权重为 3，顶点 2 与顶点 3 相连，权重为 4。

在第二个例子中，不存在这样的树 T。例如，顶点 1 与顶点 2 相连，权重为 2，而顶点 1 与顶点 3 相连，权重为 2 的树不满足条件，因为在 G 中 1 和 2 之间的最短路径是 3，而在这棵树里是 2，不相等。

问题 N. 漂亮的花束

时间限制 2 秒 内存限制 1024 兆字节

环球热带植物中心（UTPC）拥有一个种植园，从西到东直线种植了 N 棵树，从西开始从 1 到 N 编号。

种植园即将进入*为期 K 天的*收获期。在此期间，每棵树每天正好开一朵花，花色有红、绿、蓝三种。*第* i 棵树在 K 天内的开花时间表用字符串 S_i表示，其中*第* j 个字符 $S_{i,j}$表示*第* j 天开花的颜色："R "表示红色，"G "表示绿色，" B "表示蓝色。

每天，所有盛开的鲜花都会被采摘下来，组合成每束三朵花的花束。一束有效的花束必须由颜色相同或颜色不同的所有花朵组成。理想情况下，采收的鲜花应被组合成这样的花束，没有任何剩余。然而，并非总能做到这一点。

为了解决这个问题，UTPC 允许**在**采收期开始**前**对树木进行以下操作。每种操作都可以执行任意次数，但任何树木都不能被操作超过一次：

- 选择一个整数 i（$1 \leq i \leq N$），砍掉*第* i 棵树。这棵树将不会收获任何花朵。
- 选择一个整数 i（$1 \leq i \leq N$），并对第 i 棵树使用生长加速器。在这段时间内，这棵树每天会开两朵花，*在第 j 天都会*开出 $S_{i,j}$所指示颜色的花。

收获期开始后就不能再进行操作了。

由于UTPC 的办公室位于种植园的西边，因此它不希望对东边的树木进行操作。因此，一组操作的**成本**定义为**最东边被操作树木的指数**。如果没有对树木进行操作，则成本为 0。

确定最低可能成本，以确保在采收期的每一天，所有鲜花都能组合成有效的花束，且没有任何剩余。注意，如果所有树木都被砍伐，则认为没有剩余的鲜花。

输入

输入格式如下

```
N K
S(1) S(2)
.
SN
```

- N 和 K 均为整数。
- $1 \leq N, K.$
- $NK \leq 10^5$.
- $|S_i| = K$。
- S_i中的每个字符都是 "R"、"G "或 "B "中的一个。

输出

打印一个整数 - 确保采收期内任何一天都不会有花未分组的最低成本。

范例

标准输入	标准输出
4 5 rbgr bggbr rbgbr rrrrrrr	2
3 3 RGB BGG GGR	0
3 4 GGGG BGGG GGGR	3
6 4 bggb bggb rgbg rrrr gggg bbbb	3

注释

在第一个例子中，砍掉第二棵树，开出的花是

- 第 1 天：RRR
- 第 2 天：GBR
- 第 3 天：BGR
- 第 4 天：GBR
- 第 5 天：RRR

每天都可以形成花束，没有剩余。成本为 2，不可能实现成本为 1 的目标。

在第二个例子中，不需要任何操作，因此成本为 0。

在第三个例子中，必须砍伐所有树木才能满足条件。成本为 3。

在第四个例子中，通过对第 1 棵树使用生长加速器并砍掉第 3 棵树，每天的开花结果变为

- 第一天：BBBRGB
- 第二天：GGGRGB
- 第 3 天：GGGRGB

- 第 4 天：BBBRGB

这些都可以组合成有效的花束。费用为 3.

问题 O. 一个不同的不等式

时间限制 3 秒 内存限制 1024
兆字节

给你一个整数 N 和一个长度为 $N-1$ 的字符串 S，字符串 S 由< 和 > 组成。设 $P = (P_1, P_2, - - - , P_{(N)})$ 是 $(1, 2, - - -, N)$ 的排列

。

如果一个排列 P 满足以下条件，则称其为**好排列**。

- 对于每个 i（$1 \leq i \leq N-1$），如果 S 的第 i 个字符是 <，则 $P_i < P_{(i)(+1)}$；如果是 >，则 $P_i > P_{(i)(+1)}$。

如果一个排列 P 满足以下条件，则称它为**奇妙排列**。

- P 是好排列。
- 使$|P_{(i)} - P_{(i)+1}| = 1$ 的指数 i 的个数（$1 \leq i \leq N-1$）在所有好排列中是最大的。

您的任务是计算 998244353 模的奇妙排列数。

输入

输入的格式如下：

```
N
S
```

- N 是整数。
- $2 \leq N \leq 2 \times 10^5$
- S 是长度为 $N-1$ 的字符串，由< 或 > 组成。

输出

单行输出答案。

示例

标准输入	标准输出
5 <<>>	2
40 <<>><>><>>>><><<><><><<>><<<<>><><<<>><	535474657

备注

在第一个测试案例中，$(1, 2, 5, 4, 3)$ 和 $(2, 3, 5, 4, 1)$ 是良好排列。满足
$|P_i - P_{(i)(+1)}| = 1$ 的个数是 3，2。

我们可以证明，在良好排列组合中，满足$|P_{(i)} - P_{(i)(+1)}| = 1$ 的 i 的最大个数是 3，而奇妙排列组合是（1，2，5，4，3）和（3，4，5，2，1）。

问题 P. 树上的完美绥卡游戏

时间限制 5 秒 内存限制 1024 兆字节

给你一棵树 T，树上有 N 个顶点，标签从 1 到 N。第 i 条边连接顶点 u_i和 v_i。

每个顶点都有一个称为**级别**的正整数。最初，顶点 $v = 1, 2, . . , N$ 为 A_v。

我们在树 T 上考虑以下问题：

> 判断是否有可能通过执行以下操作 $N - 1$ 次，将树 T 变成一棵只有一个顶点的树：
>
> - 选择一条端点等级相同的边，并将其收缩。设 l 为两个端点的共同等级，则收缩后的新顶点的等级为 $l + 1$。

您需要处理 Q 个查询。在第 i 个查询中，您得到了边的编号 e_i。交换顶点 $u_{(e)(i)}$和 $v_{(e)(i)}$在树 T 中的级别（这种交换也会影响所有后续查询）后，输出上述问题的答案。

输入

输入的格式如下

```
N
u(1) v(1)
u(2) v(2)
.
uN-1 vN-1 A(1) A(2).
.A_N Q
e(1)
e(2)
.
.
e_Q
```

- 所有输入值均为整数。
- $2 \leq n \leq 2 \times 10^5$.
- $1 \leq u_i, v_i \leq N$.
- $1 \leq A_i \leq N$.
- $1 \leq q \leq 2 \times 10^5$.
- $1 \leq e_i \leq N - 1$.
- 给定图形是一棵树。

输出

输出 Q 行。对于第 i 个查询，在交换顶点 $u_{(e)(i)}$和 $v_{(e)(i)}$的层级后，如果可以使用上述操作将 T 转化为单顶点树，则输出 "是"；否则，输出 "否"。

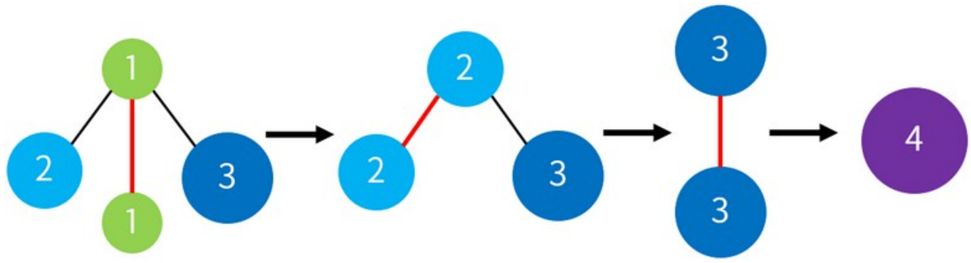
示例

标准输入	标准输出
4 1 2 1 3 1 4 1 1 2 3 4 1 2 3 1	是 否 否 是
20 1 2 1 3 2 4 1 5 2 6 5 7 4 8 3 9 6 10 7 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 4 4 7 3 8 2 8 6 4 2 3 3 4 5 6 5 4 3 3 6 10 8 19 5 9 19 10 19 19 10 19	无 否 否 否 是 否 否 否 否 是 否

注释

在第一个示例的第一个查询中，交换顶点 $u_1= 1$ 和 $v_1= 2$ 的级别后，顶点 1、2、3、4 的级别分别变为 1、1、2、3。在这种情况下，可以进行操作（选择合适的边），使树变成一个级别为 4 的单顶点。因此，输出结果是 "是"。下图可能对您有所帮助。

在第二个查询中，交换顶点 $u_2= 1$ 和 $v_2= 3$ 的级别后，顶点 1、2、3、4 的级别分别为



Рис。 1: 第一个测试案例中第一个查询的图示

分别变为 2、1、1、3。在这种情况下，根本无法执行任何操作，也不可能将树转化为单一顶点。因此，输出为 "否"。

问题 Q. 二次方块

时间限制 2 秒 内存限制 1024
兆字节

给您一个整数序列 $A=(A_1, A_2, ..., A_N)$ 长度为 N 。

连续子序列 $(A_L, A_{(L)(+1)}, .., A_R)$ 由整数 L 和 R 定义，且 $1\leq L\leq R\leq N$ ，如果满足以下条件，则称其为**二次子序列**：

- 存在实数 a、b、c，使得对于满足 $L\leq i\leq R$ 的每个整数 i，方程
  $A_i= ai^2+ bi+ c$ 成立。

你的任务是将序列 A 分割成几个连续的**二次**子序列。在所有可能的方法中，输出最少的子序列数。

给你 T 个测试案例。输出每个测试用例的答案。

输入

输入的格式如下

T 案例$_1$案 例$_2$. 案例$_T$

每个测试用例的格式如下：

N $A_{(1)}$ A(2)‧ ‧$A_{(N)}$个

- 所有输入均为整数。
- $1\leq T\leq 10^5$.
- $1\leq N\leq 2\times 10^5$.
- $-10^{18}\leq A_i\leq 10^{18}$.
- 在单个输入的所有测试用例中，N 的总和最多为 2×10^5.

输出

输出 T 行。
对于第 i 行，打印第 i 个测试用例的答案。

示例

标准输入	标准输出
4 12 -16 -9 -4 -1 0 0 0 0 1 4 9 16 8 2 0 2 5 0 3 0 8 1 0 5 1000000000000000000 250000000000000000 0 250000000000000000 1000000000000000000	3 3 1 1

备注

在第一个例子中，给定序列可以划分为三个二次子序列：$(-16, -9, -4, -1)$, $(0, 0, 0)$, 和 $(0, 1, 4, 9, 16)$。对于每个子序列，$(a, b, c) = (-1, 10, -25)$、$(0, 0, 0)$和$(1, -16, 64)$都满足条件。不可能将序列分成少于 3 个二次子序列，所以答案是 3。

在第四个示例中，请注意输入值可能超出 32 位整数范围。