订阅DeepL Pro以翻译大型文件。
欲了解更多信息，请访问www.DeepL.com/pro。

第三届环球杯

第 4 阶段：Hong¯o

2024 年 7 月 13-14 日

这套试题应包含 17 个问题，共 28 页。

问题 A.再次使UTPC

输入文件：　标准输入
输出文件：　标准输出
时间限制　2 秒 内存限制
1024 兆字节

给你一个长度为 N 的字符串 S。S 中的每个字母都是 U、T、P 或 C：

- 选择一对符合 1 的整数 (i, j) $\leq i \leq j \leq N$.按字母升序从 S 的第 i 个字母到第 j 个字母排序。

找出满足以下条件的可能性，并计算出可能的最小操作次数。

- S 包括作为连续子串的 UTPC。

您需要解决 T 个测试案例。

输入

输入信息来自标准输入，格式如下，其中$_i$ 代表第 i 个测试用例：

```
T
1 案
例2
.
个案T
```

每个案例的格式如下

```
N
S
```

- T、N 均为整数。
- $1 \leq T \leq 2 \times 10^5$
- $1 \leq N \leq 2 \times 10^5$
- S 是一个字符串，由 U、T、P 和 C 组成，长度为 N。
- 对于每个输入文件，所有测试用例的 N 之和不超过 2×10^5 。

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，如果可以满足条件，则打印最小操作数。如果不可能，则打印-1。

示例

标准输入	标准输出
3 10 UCUCTPUCUC 5 UTCUP 12 TUPCTTPCUTPC	2 -1 0

备注

对于第一个测试用例，可以通过以下两个操作来满足条件。一次操作无法满足条件。

- 选择 $(i, j) = (1, 4)$。S 变为 CCUUTPUCUC。
- 选择 $(i, j) = (7, 10)$。S 变为 CCUUTPCCUU。

对于第二个测试案例，无法满足条件。

对于第三个测试用例，操作不是满足条件的必要条件。

问题 B. 黑或白 2

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

给你一个整数 N、M 和 K，N 和 M 都大于或等于 2。

在 N 行 M 列的网格中，K 个单元为黑色，其余 $NM - K$ 个单元为白色。这里，彩色网格

的折线定义如下：

- 包含 2 个黑色单元格和 2 个白色单元格的 2×2 子网格的数量。

请提供一种给网格上色的方法，使**折腾**最小化。

您将收到 T 个测试用例，您需要为每个测试用例提供一个解决方案。

输入

输入信息来自标准输入，格式如下，其中$_i$ 代表*第* i 个测试用例：

```
T
₁案
例₂
.
个案_T
```

每个测试用例的格式如下

```
N M K
```

- 所有输入值均为整数。
- $1 \le T \le 10^5$
- $2 \le n, m \le 1500$
- $0 \le K \le NM$
- 对于每个输入文件，所有测试用例的 NM 总和不超过 4×10^6 。

输出

按顺序输出每个测试用例的答案，以行分隔。

对于每个测试用例，输出一个长度为 M 的字符串，由 N 行中的 0 和 1 组成。

如果第 i 行输出的字符串的第 j 个字符为 0，则表示从上往左数第 i 个、第 j 个的方格被涂成白色，1 表示从上往左数第 i 个、第 j 个的方格被涂成黑色。

如果有不止一种以最小**抛掷**值填满方格的方法，则输出其中一种。注意不要输出最小**抛掷**值。

示例

标准输入	标准输出
2 2 2 2 2 3 0	10 01 000 000

备注

- 在第一个测试案例中，**抛掷**的最小值是 1。
- 对于第二个测试案例，**抛掷**值为 0，也就是最小值。

问题 C. 等值线乘法

输入文件：	标准输入
输出文件：	标准输出
时间限制	3 秒 内存限制
	1024 兆字节

有一个长度为 2 的序列^N，由 $A_0, A_1, \ldots$ 给出。$, A\ N_{2-1}$ 。最初，$A_0 = A_1 = \cdot\ \cdot\ \cdot = A\ N_{2-1} = 1$ 。

您将执行 K 次操作。在第 i 次操作中，对于每个 j（其中 $0 \le j < 2^N$），如果 $\mathrm{popcount}(j \oplus C_i) = D_i$，将 A_j 替换为 $(A_j \times X_i) \bmod M$。

确定 $A_0, A_1, \ldots$ 的值。$, A\ N_{2-1}$ 进行所有运算后的值。

输入

输入内容由标准输入法提供，格式如下

$N\ M\ K$
$C_1\ D_1\ X_1$
$C_2\ D_2\ X_2$
.
$C_M\ D_M\ X_M$

- 所有输入值均为整数。
- $1 \le N \le 18$
- $2 \le M \le 10^9$
- $1 \le K \le 5 \times 10^5$
- $0 \le C_i < 2^N$
- $1 \le D_i \le N$
- $2 \le X_i \le 10^9$

输出

执行所有操作后，输出 $A_0, A_1, \ldots$ 的值。$, A\ N_{2-1}$ 在执行完所有操作后，用空格分隔成一行。

实例

标准输入	标准输出
3 100 2 0 2 4 3 0 25	1 1 1 0 1 4 4 1

4 998244353 7 0 2 4 3 0 25 9 4 37 4 1 16 6 3 8 1 4 68 13 3 97	1552 8 1 9700 1 64 229696 1 8 4 388 8 64 8 68 1

问题 D. DRD 字符串

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

如果存在特定的非空字符串 D 和 R，且 S 是由 D、R、D 按此顺序连接而成，则字符串 S 称为 **DRD** 字符串。现在我们可以使用 M 种字母。求长度为 N 的 DRD 字符串的数目（模为 998244353）。

输入

输入内容由标准输入法提供，格式如下

N M

- 所有输入值均为整数。
- $3 \leq N \leq 10^6$
- $1 \leq M \leq 10^6$

输出

单行打印答案。

实例

标准输入	标准输出
6 2	40
3017 7801	515391664

备注

如果使用例 1 中的 a 和 b，abbaab 和 aaaaaa 是长度为 6 的 DRD 字符串，但 abbabb 和 aaabbb 不是。

问题 E. 平分

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

给定两个整数 N 和 M。您必须在 $N \times M$ 的网格中，每个单元格都要填入一个从 1 到 NM 的整数，每个整数都要出现一次。

如果符合以下条件，有效填充就被定义为**公平拟合**：

- 从 1 到 NM 的每一个整数都会在其中一个单元格中精确写入一次。
- 每行写入的 M 个整数之和在所有行中都是相同的。

确定是否存在这种**公平匹配**，如果存在，请举例说明。

您将收到 T 个测试用例，您需要为每个测试用例提供一个解决方案。

输入

输入信息来自标准输入，格式如下，其中$_i$ 代表第 i 个测试用例：

```
T
₁案
例₂
.
个案T
```

每个测试用例的格式如下

```
N M
```

- 所有输入值均为整数。
- $1 \leq T \leq 10^4$
- $1 \leq N, M$
- $1 \leq nm \leq 3 \times 10^5$
- 对于每个输入文件，所有测试用例的 NM 总和不超过 5×10^5 。

输出

按顺序输出每个测试用例的答案，行与行分开。对于每个

测试用例，如果不存在**公平拟合**，则输出 "否"。

否则，请按以下格式输出一个**公平拟合**的示例：

```
是
S1,1 S1,2 ··· ·S1,M
S2,1 S2,2 · ·S2,M

.
SN,1 SN,2 ·SN,M
```

这里，$S_{i,j}$ 表示写在从上往下第 i 个、从左往下第 j 个方格中的整数。

示例

标准输入	标准输出
2 2 2 10 1	是 1 4 2 3 没有

备注

- 对于第一个测试用例，每行写入的 M 个整数之和为 $5 = 1 + 4 = 2 + 3$。
- 对于第二个测试案例，可以证明不存在**公平拟合**。

问题 F. 翻转与否

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

一排有 N 张牌。最初，如果字符串 S 的*第* i 个字符为 1，则从左边开始的*第* i 张牌正面朝上；如果字符串 S *的第* i 个字符为 0，则从左边开始的*第* i 张牌正面朝下。您最多可以执行以下操作 10^6 次：

- 将最右边的牌移动到最左边的位置。如果被移动的牌是正面朝上的，则从左边翻开位置 A_1 , A_2 , ... 的所有牌。, A_P 。此外，您还可以选择从*左边翻*开位置 B_1 , B_2 , ... 的所有牌，�� *者从左边翻开位置* B 的所有牌。, B_Q 的所有纸牌，或者什么也不做。

经过这些操作后，如果字符串 T 的*第* i 个字符为 1，则希望从左边开始的*第* i 张牌面朝上，如果字符串 T *的第* i 个字符为 0，则希望从左边开始的*第* i 张牌面朝下。请确定是否可以用 10^6 或更少的操作来满足该条件，如果可以，请输出一个操作序列，以最少的操作次数来满足该条件。

输入

输入以下列格式从标准输入端提供：

```
N
S
T
P
A1 A2 . .AP Q
B1 B2 . .BQ
```

- 所有数字输入均为整数。
- $1 \le N \le 5000$
- S 和 T 都是长度为 N 的字符串，由 0 和 1 组成。
- $S \neq T$
- $1 \le P, Q \le N$
- $1 \le a_1 < a_2 < \cdots < a_P \le n$
- $1 \le b_1 < b_2 < \cdots < b_Q \le n$

输出

如果无法用 10^6 或更少的操作满足条件，则输出-1。如果可以满足条件，则按以下格式输出一个操作次数最少的操作序列：

```
M
U
```

这里，M 是操作次数，U 是长度为 M 的字符串，仅由 0 和 1 组成，代表操作序列。如果 U 的第 i 个字符是 1，则表示在*第* i *次*操作中，从左边开始的 B_1 , B_2 , ... *的*所有纸牌都被翻转。, B_Q 的所有纸牌都被翻转。如果字符为 0，则不进行任何操作。

实例

标准输入	标准输出
5 00001 00111 3 1 2 3 2 3 5	4 1001
4 0110 1000 2 1 2 4 1 2 3 4	-1

备注

对于第一种情况，在第一次操作中，将最右边的一张牌移动到最左边，这张牌的状态就会变为 10000，因为被移动的牌是朝上的，所以我们翻转左起位置 A_1, A_2, A_3 （第 1，2，3）的牌，结果是 01100。接下来，如果我们选择翻转位于左起 B_1, B_2 （第 3、5）位置的纸牌的正面/背面，则状态变为 01001。继续按照输出示例，在第二次操作中，状态变为 01000，在第三次操作中，状态变为 00100，在第四次操作中，状态变为 00111。没有任何方法可以在更少的操作中实现这一点，因此输出示例是正确的输出。

对于第二个测试案例，无法在 10^6 次操作内满足条件。

问题 G. 图表加权

输入文件：	标准输入
输出文件：	标准输出
时间限制	5 秒 内存限制
	1024 兆字节

有一个连通的无向图，图中有 N 个顶点，编号为 $1, 2 ., N$ 和 M 条边。第 i 条边连接顶点 u_i 和顶点 v_i 。该图可能包含同一对顶点之间的多条边，但不包含自循环。

对于每个 $W = 0, 1, ., K$，求解以下问题

确定是否存在为每条 i 边分配权重 w_i $0, 1, ., L$ 给第 i 条边，每 i 条边的权重为 $1 \leq i \leq M$，这样图中任何一棵生成树的权重都正好是 W。生成树的权重定义为生成树中所有边的权重之和。如果存在这样的分配，请找出 $(w_1)^2 + (w_2)^2 + \cdots + (w_M)^2$ 在所有此类分配中的最小值。

输入

输入内容由标准输入法提供，格式如下

$$
\begin{array}{l}
N\ M\ K\ L \\
u_1\ v_1 \\
. \\
u_M\ v_M
\end{array}
$$

- 所有输入值均为整数。
- $2 \leq N \leq 10^5$
- $n - 1 \leq m \leq 2 \times 10^5$
- $1 \leq L, K \leq 10^5$
- $1 \leq u_i, v_i \leq N$
- $u_i \neq v_i$
- 给定的无向图是连通的。

输出

对于每个 $W = 0, 1, ., K$，按此顺序输出问题答案，中间用空格隔开。具体来说，如果没有符合条件的赋值，则输出-1。如果存在赋值，则输出 $(w_1)^2 + (w_2)^2 + \cdots + (w_M)^2$ 在所有此类赋值中的最小值。

实例

标准输入	标准输出
4 4 3 2 1 2 2 3 2 4 3 4	0 1 3 4
2 3 2 1 1 2 2 1 1 2	0 3 -1
6 7 9 2 1 2 2 3 2 4 4 5 4 6 1 4 3 4	0 1 2 5 6 7 10 13 22 25

备注

例如，在例 1 中，当 $W = 2$ 时，如果我们设置 $(w1, w2, w3, w4) = (0, 1, 1, 1)$，则图中任何生成树的权重都是 2。

在例 2 中，不可能让图中任何生成树的权重等于 2。

问题 H. 巨大的分段树

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

如果以下条件成立，则认为一个整数区间是*段树状*的：

- 可表示为 $[2^i j, 2^i (j+1))$ 的区间（$0 \le i \le K, 0 \le j < 2^{K-i}$），其中有一些整数 i, j.

对于满足 $0\ l < r\ 2^K$ *的*一对整数 (l, r) ，可以证明区间 $[l, r)$ 可以表示为段树状区间的联合。我们用 $f(l, r)$ 表示所需的最小区间数。

对于 $k = 1, 2, .., 2K - 2$，求解下列问题：

- 求使 $f(l, r) = k$ 的整数对 (l, r) $(0 \le l < r \le 2^K)$ 的个数，模为 998244353.

输入

输入内容由标准输入法提供，格式如下

```
K
```

- K 是一个整数。
- $2 \le K \le 5 \times 10^5$

输出

当 $k = 1, 2, ...$ 时，*依次*打印问题答案。, $2K - 2$ 时的答案。

实例

标准输入	标准输出
3	15 14 6 1
5	63 110 132 114 70 30 8 1
10	2047 4975 10896 21772 38360 58724 77184 86312 81448 64324 42112 22576 9744 3304 848 155 18 1

备注

在第一个例子中，当 $k = 4$ 时，$f(l, r) = k$ 仅在 $l = 1$、$r = 7$ 时成立，因此 1 是输出结果。

问题 I. 我爱马拉松比赛

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

马拉松比赛将在一个池塘边举行。池塘是圆形的，沿周长顺时针等间隔标有 1、2、......、2N。, $2N$ 沿顺时针方向间隔相等。共有 $2N$ 人参加马拉松比赛，其中 N 人戴红帽子，其余 N 人戴白帽。

马拉松比赛的流程如下：

- 对于 $2N$ 个标记中的每个标记，正好有一名参赛者占据该位置。
- 在标有 1 的位置上的参赛者接过接力棒，开始顺时针跑。
- *第* i 个（$1 \leq i \leq 2N-1$）选手继续跑，直到跑到第一个戴着与他/她不同颜色帽子的人的位置。到达该位置后，他/她将接力棒传给那个人，然后离开池塘。接过接力棒的人开始顺时针跑。
- 第 $2N$ 位选手跑到标有 1 的位置，完成马拉松。

如果绕池塘一圈的长度是 1，那么 $2N$ 名参赛者所跑距离的总和就是一个整数，即 L。

有 $(2N)!$ $2N$ 个参与者的可能排列。求模为 998244353 的所有参与者的 L 之和。

输入

从标准输入端输入的数据格式如下

```
N
```

- 所有输入的数字都是整数。
- $1 \leq N \leq 10^6$

输出

将答案打印在一行上。

实例

标准输入	标准输出
1	2

2	40
3	1656
4	112896
5	11750400

备注

在第一个例子中，有两种可能的排列方式，两种情况下 $L = 1$。

在第二个例子中，如果在标有

1、2、3、4 是

- 红、红、白、白，则 $L = 2$。
- 红、白、红、白，则 $L = 1$。
- 红、白、白、红，则 $L = 2$。
- 白、红、红、白，则 $L = 2$。
- 白、红、白、红，则 $L = 1$。
- 白、白、红、红，则 $L = 2$。

每个参与者都有 4 种可能的安排。所有 24 种安排的 L 之和为（$2 + 1 + 2 + 2 + 1 + 2$）$\times 4 = 40$。

问题 J. 日本礼金

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

有 N 种纸币，第 i 种纸币是 A_i -日元纸币。每种纸币有 10 张^{100} 。这里，$A_1 < A_2 < \cdots < A_N$ 成立，对于 $1 \le i \le N-1$ 的每个 i，A_{i+1} 是 A_i 的倍数。

你要从这些纸币中选出一些，装进一个信封。

如果满足以下条件，把纸币装进信封的方法就叫做**装** x **日元**的**好方法**：

- 信封里的总金额是 x 日元。
- 不可能从信封中挑选出总金额正好为 $\frac{x}{2}$ 日元的纸币。

此外，如果有一种把 x 日元放进去**的**好方法，那么 x 日元就被称为好**金额**。求介于 L 日元和 R 日元（含 R 日元）之间的好金额的个数。

输入

输入以下列格式从标准输入端提供：

```
N L R
A_1 A_2 ... A_N
```

- 所有输入均为整数。
- $1 \le N \le 60$
- $1 \le L \le R \le 10^{18}$
- $1 = a_1 < a_2 < \cdots < a_N \le 10^{18}$
- A_{i+1} 是 A_i 的倍数 $(1 \le i \le N-1)$

输出

将答案打印在一行上。

实例

标准输入	标准输出
3 20 30 1 5 10	8

8 5000074846028445443 985892611352151235 1 1971 151767 10927224 87417792 118975614912 263174060185344 43686893990767104	483957600323779237

备注

例如，30 日元是一个很好的金额，因为放三张 10 日元的纸币就可以放 30 日元。

另一方面，20 日元并不是一笔好钱，因为没有好办法把 20
日元。

在 20 日元和 30 日元之间有 8 个不错的金额：21、23、25、26、27、28、29 *和* 30 日元。

问题 K. Kth Sum

输入文件： 标准输入
输出文件： 标准输出
时间限制 3 秒 内存限制
1024 兆字节

给你三个整数序列 $A = (A_1, A_2, \ldots, A_N)$，$B = (B_1, B_2, \ldots, B_N)$, 和 $C = (C_1, C_2, \ldots, C_N)$，每个长度为 N。

考虑所有可能的和，其形式为 $A_i + B_j + C_k$，其中 $1 \le i, j, k \le N$.你的任务是在这 N 个^3 和中找出 K 个最小的和。

输入

第一行包含两个整数 N 和 K（$1 \le N \le 50,000, 1 \le K \le \min(N^3, 10^9)$）。第

二行包含 N 个整数 $A_1, A_2, \ldots, a_N$ （$0 \le a_i \le 10^9$）。

第三行包含 N 个整数 $B_1, B_2, \ldots, b_N$ （$0 \le b_j \le 10^9$）。第

四行包含 N 个整数 $C_1, C_2, \ldots, c_N$ （$0 \le c_k \le 10^9$）。

输出

打印 $A_i + B_j + C_k$ 形式的所有可能和中的第 K 个最小和。

实例

标准输入	标准输出
2 4 1 2 3 4 5 6	10
10 40 11 9 13 12 15 11 11 2 11 17 3 1 10 2 12 18 9 11 11 15 14 9 4 14 16 9 20 2 1 18	14
1 1 1000000000 1000000000 1000000000	3000000000

备注

对于第一个测试用例，所有可能的和都是 9、10、10、10、11、11、11、12，按升序排列。因此

第 4 个最小和是 10。

问题 L.最大三角形

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

有 N 根直木棍。第 i 根木棒的长度为 L_i。

考虑制作一个（非退化）三角形，在其中选取 3 根木棒。确定选择是否存在，如果存在，计算可制作三角形最大表面的平方。

有 T 个测试案例，请逐一回答。

输入

输入内容按以下格式从 "标准输入 "中输入。其中，$_i$ 表示第 i 个测试用例。

```
T
₁案
例₂
.
个案T
```

每个测试用例的格式如下

N
$L_1\ L_2\ ..L_N$

- 所有输入均为整数。
- $1 \leq T \leq 2\times 10^5$
- $3 \leq N \leq 3\times 10^5$
- $2 \leq L_i \leq 20000$
- L_i 是偶数。
- 在每个输入中，所有测试用例的 N 之和等于或小于 2×10^5。

输出

打印 T 行。在第 i 行，打印第 i 个测试用例的答案。

在每个测试用例中，如果没有有效的 3 根小棒，则打印-1。如果存在，则打印三角形最大面积

的平方。注意，该值在约束条件下为整数。

示例

标准输入	标准输出
3 5 2 2 2 2 2 7 2 6 4 10 8 10 20 5 4 16 36 64 100	3 1344 -1

备注

在第二个测试案例中，三边长分别为 8、10、10 *的*三角形的最大面积为 $8\sqrt{21}$。 —
是 1344。

在第 3 个测试案例中，没有三个条形不能组合成非退化三角形。

问题 M. 多数和排列

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

给你一个整数序列 $(A_1, A_2, \ldots, A_M)$ ，由 1 到 $2N$ 之间的奇数组成 ‐ 1
(包括在内）。

求满足以下条件的（$1, 2, \ldots, 2N$ ）的排列组合 $P = (P_1, P_2, \ldots, P_{2N})$ 的个数，模数为 998244353
。

- 存在一个长度为 $2N$ 的二进制字符串 S，它只由 0 和 1 组成，且满足以下所有条件：
  - S 中 0 和 1 的频率正好各为 N。
  - 对于每个 $i = 1, 2, ., M$, 在 $1, 2, ., A_i$ -th 字符中，出现频率最高的字符是 0.
  - 对于每个 $i = 1, 2, ., M$, 在 $P_1, P_2, ., P_{A_i}$ -th 字符中，出现频率最高的字符是 0。

输入

输入以下列格式从标准输入端提供：

$N\ M$
$A_1\ A_2\ .\ .A_M$

- 所有输入均为整数。
- $1 \le M \le N \le 10^5$
- $1 \le a_1 < a_2 < \cdots < a_M \le 2n - 1$
- $_i$ 很奇怪。

输出

将答案打印在一行上。

示例

标准输入	标准输出
2 2 1 3	14

备注

例如，如果 $P = (2，1，3，4)$，那么 $S = 0011$ 满足所有三个条件。

另一方面，如果 $P = (4，3，2，1)$，则不存在满足所有三个条件的长度为 4 *的*字符串。

问题 N. 缩略语数量

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

你有一个字符串 $S = S_1 S_2 \ldots S_N$，长度为 N，由小写英文字母组成。请计算精确执行一次下面的操作可以得到多少个不同的字符串：

- 选择整数 l 和 r，使得 $1 \leq l \leq r \leq N$，并删除从第 1 个到第
  结果字符串为 $S_1 S_2 \ldots S_{l-1} S_{r+1} \ldots S_N$.

输入

输入以下列格式从标准输入端提供：

```
N
S
```

- N 是一个整数。
- $1 \leq N \leq 5 \times 10^5$
- S 是一个长度为 N 的字符串，由小写英文字母组成。

输出

将答案打印在一行上。

实例

标准输入	标准输出
5 abb	11
5 aaaaa	5
4 utpc	10

备注

在第一个例子中，可能产生的字符串有以下 11 种：

- 空字符串
- a
- aab
- ab
- abab
- 阿布

- 阿爸
- 减退
- b
- 狒狒
- bbab

问题 O. 优化列车运行

输入文件： 标准输入
输出文件： 标准输出
时间限制 3 秒 内存限制
1024 兆字节

UTPC铁路沿线有N+1个车站，从起点站到终点站从0到N连续编号。对于每个 i（$0 \le i \le N-1$），车站 i 和车站 $i+1$ 相邻，这些车站之间的拥挤程度为 C_i。目前，"铁路货场 "位于车站 0 和 N。

在下一次修订时刻表时，将通过多次重复以下操作来建造铁路货场：

- 选择一个车站 i（$1 \le i \le N-1$），并在此修建一个铁路货场。这项业务的成本为 A_i。

接下来，列车将通过多次重复以下操作在铁路站场之间运行：

- 选择铁路站场所在的 l 站和 r 站 （$l < r$），在这两个站之间运行一列列车。这一操作会使 i 站和 $i+1$ 站（$l \le i < r$）之间的拥堵程度降低 1。

修订时刻表的目标是将 i 站和 $i+1$ 站（$0 \le i \le N-1$）之间的拥堵程度降至 0 或以下。请计算为实现这一目标，建造铁路站场和运营列车所需的最低总成本。

输入

输入内容由标准输入法提供，格式如下

```
N
C0 C1 ... CN-1
A1 A1 ... AN-1
```

- 所有输入值均为整数。
- $2 \le N \le 5 \times 10^5$
- $1 \le C_i \le 10^9$ $(0 \le i \le N-1)$
- $1 \le A_i \le 10^9$ $(1 \le i \le N-1)$

输出

输出答案。

实例

标准输入	标准输出
4 3 1 4 1 5 9 2	15
9 28 35 19 27 84 98 78 79 60 40 35 54 63 72 71 27 94	682

备注

在第一个例子中，在第 3 车站建立了一个铁路货场，在第 0 和第 3 车站之间设置了 3 列火车，在第 0 和第 4 车站之间设置了 1 列火车。结果，每个区段的拥堵程度都变为 0 或更低，总成本为 15 。这就是最低成本。

问题 P. 优先队列 3

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

给你一个长度为 $N+M$ 的字符串 S，由 $N+$ 个字符和 M - 个字符组成，以及一组 $a = \{a_1 , a_2 , ..., A_M\}$ 由 M 个整数组成。

准备两个集合 $X = \{\}$ 和 $Y = \{\}$，对 $i = 1, 2, ., N+M$：

- 当 S 的第 i 个字符为 "+"时，从 1 到 N 中选择一个既不包含在 X 也不包含在 N 的整数。或 Y，并将其添加到 X 中。
- 当 S 的第 i 个字符为"-"时，从 X 中删除 X 中包含的最小整数 m，并加上 m 到 Y。根据约束条件，可以保证在执行此操作前 X 不为空。

有 N 种方法可以确定要加到 X 中的整数的顺序，请找出其中有多少种方法在进行所有运算后，$Y = A$。

输入

从标准输入端输入的数据格式如下

```
N M
S
A1 A2 . .AM
```

- 所有输入的数字都是整数。
- $1 \le m \le n \le 300$
- S 是长度为 $N+M$ 的字符串，由 $N+$ 个字符和 M - 个字符组成。
- 对于 $i = 1$、2、......, $N+M$ 时，第 i 个字符之前出现的 - 字符数不超过第 i 个字符之前出现的 + 字符数。
- $1 \le a_1 < a_2 < \cdot\cdot\cdot < a_M \le n$

输出

将答案打印在一行上。

实例

标准输入	标准输出
4 2 ++-++- 1 3	4
6 4 ++-++---++ 2 3 4 6	48
20 10 ++++-++++++--+--+-+++++--+-++- 1 2 3 4 5 6 7 9 12 13	179396825

备注

在第一个例子中，作为满足条件的操作序列的示例，可以考虑以下操作：

- 现在 $X = \{3\}$，$Y = \{\}$。
- 现在 $X = \{3, 4\}$，$Y = \{\}$。
- 当 $i = 3$ 时，从 X 中删除最小整数 3 并将其添加到 Y 中。现在 $X = \{4\}$，$Y = \{3\}$。
- 现在 $X = \{2, 4\}$，$Y = \{3\}$。
- 现在 $X = \{1，2，4\}$，$Y = \{3\}$。
- 当 $i = 6$ 时，从 X 中删除最小整数 1 并将其添加到 Y 中。现在 $X = \{2，4\}$，$Y = \{1，3\}$。

在第二个例子中，S 的末尾不一定是"-"。

问题 Q. 商和

输入文件：	标准输入
输出文件：	标准输出
时间限制	2 秒 内存限制
	1024 兆字节

给你一个由 N 个不同的正整数组成的序列 $A = (A_1, A_2, \ldots, A_N)$。考虑重新排列 A 的元素，得到序列 $B = (B_1, B_2, \ldots, B_N)$。求下面表达式的最小值：

$$\left\lfloor \frac{B_1}{B_2} \right\rfloor + \left\lfloor \frac{B_2}{B_3} \right\rfloor + \cdots + \left\lfloor \frac{B_{N-1}}{B_N} \right\rfloor + \left\lfloor \frac{B_N}{B_1} \right\rfloor.$$

这里，$\lfloor x \rfloor$表示小于或等于实数 x 的最大整数。

输入

输入以下列格式从标准输入端提供：

```
N
A_1 A_2 . .A_N
```

- 所有输入均为整数。
- $2 \le N \le 2 \times 10^5$
- $1 \le A_i \le 10^{18}$
- $A_i \neq A_j \ (i \neq j)$

输出

将答案打印在一行上。

实例

标准输入	标准输出
3 2 3 6	3
2 15 4	3
9 284791808 107902 13660981249408 4622332661 13405199 24590921 361 244448137 16077087227955422	4580

备注

在第一个例子中，如果我们设 $(B_1, B_2, B_3) = (6, 2, 3)$, 则有

$$\left\lfloor \frac{b_1}{b_2} \right\rfloor + \left\lfloor \frac{b_2}{b_3} \right\rfloor + \left\lfloor \frac{b_3}{b_1} \right\rfloor = \left\lfloor \frac{6}{2} \right\rfloor + \left\lfloor \frac{2}{3} \right\rfloor + \left\lfloor \frac{3}{6} \right\rfloor = 0 + 1 + 2 = 3。$$

$B\ B\ B\ 6_{12}$ $_{3}$ 2 3