郑州轻工业大学“筑梯杯”
第十八届程序设计大赛暨省内高校邀请赛

正式赛

2026 年 4 月 6 日

在比赛开始前，请不要翻阅试题册。

Please DO NOT open the problem set before the contest starts.

主办单位

郑州轻工业大学

承办单位

郑州轻工业大学

题目概况

题号	题目名	时间限制	空间限制
A	有人截图了!	1 s	64 MB
B	保卫萝卜	2 s	512 MB
C	等差数列	1 s	512 MB
D	卖货	1 s	512 MB
E	和异位	2 s	512 MB
F	覆盖面积	3 s	512 MB
G	冰淇淋	1 s	64 MB
H	合理分配	1 s	512 MB
I	妙不可言	2 s	512 MB
J	骰子	1 s	64 MB
K	巴啦啦能量	1 s	64 MB
L	贪吃的猪	1 s	512 MB

本试题册共 12 题，18 页。

如果您的试题册不完整，请立即通知志愿者。

Problem A. 有人截图了!

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memery limit:	64 megabytes

小汪的屏幕上有一条线段，由两个端点 $A(a_x, a_y)$ 和 $B(b_x, b_y)$ 确定。他想截取屏幕的一部分，截屏区域是一个轴对齐的矩形，由两个对角顶点 (x_1, y_1) 和 (x_2, y_2) 给定（注意这两个点不一定为左上角和右下角，且坐标顺序可能任意）。

请你计算该线段落在矩形内部（含边界）的部分的长度。

Input

第一行包含四个整数 a_x, a_y, b_x, b_y，表示线段两个端点的坐标。

第二行包含四个整数 x_1, y_1, x_2, y_2，表示矩形两个对角顶点的坐标。

注意：$x_1 \neq x_2, y_1 \neq y_2$，且所有坐标的绝对值均不超过 10^4。

Output

输出一行一个实数，表示线段被矩形截取的长度。

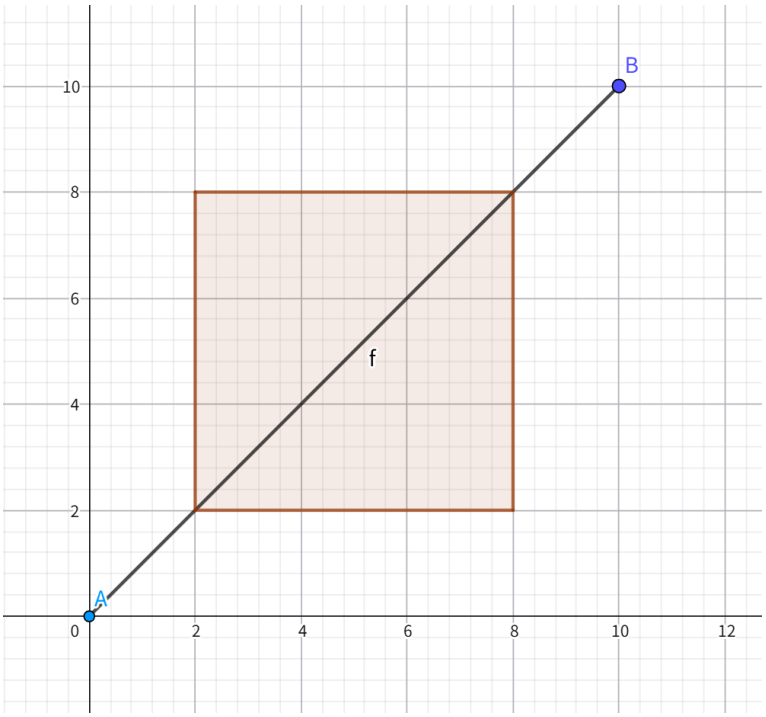
当且仅当你的答案和正确答案的绝对误差或相对误差不超过 10^{-4} 时，你的答案会被视为正确。即假设你的答案为 a，正确答案为 b，当且仅当 $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-4}$ 时，你的答案会被视为正确答案。

Example

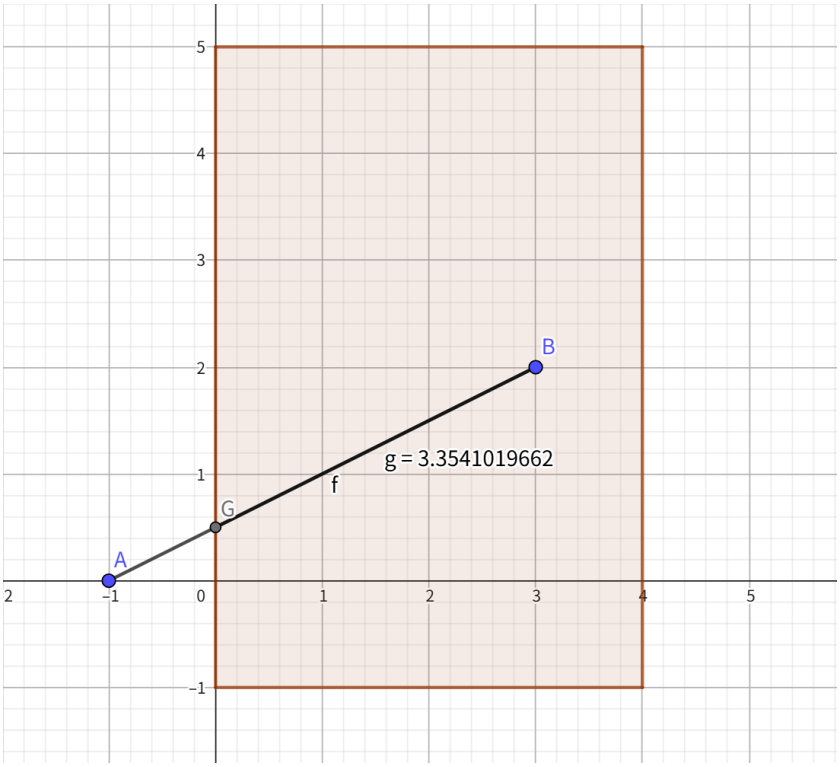
standard input	standard output
0 0 10 10 2 2 8 8	8.48528137
-1 0 3 2 4 -1 0 5	3.35410197
0 0 0 1 0 0 2 2	1.00000000

Note

在样例 1 中，线段从 $(0,0)$ 到 $(10,10)$，矩形由顶点 $(2,2)$ 和 $(8,8)$ 确定。线段在矩形内的部分是从 $(2,2)$ 到 $(8,8)$ 的线段，长度为 $6\sqrt{2} \approx 8.485281$。如图：



在样例 2 中，线段从 $(-1,0)$ 到 $(3,2)$，矩形由顶点 $(4,-1)$ 和 $(0,5)$ 确定。线段在矩形内的部分是从 $(0,0.5)$ 到 $(3,2)$ 的线段，长度约为 3.354102。如图：



Problem B. 保卫萝卜

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memery limit:	512 megabytes

在一片田野上，有 n 个庄园，按海拔**从高到低**编号为 $1, 2, \ldots, n$（编号越小海拔越高）。庄园间建有若干水道，水只能从高海拔流向低海拔，且保证从 1 号庄园（最高点）出发，水可到达所有其他庄园。

1 号庄园种有一个珍贵萝卜。外星人在某些庄园投放兔子，兔子只会**逆流而上**，目标是到达 1 号庄园吃掉萝卜。小赵可以在一个庄园设置拦截点，兔子一旦经过该点即被抓住。兔子可能选择多条逆流路径，小赵希望**无论兔子走哪条路径，都会被拦截**。换言之，拦截点必须位于**每一只兔子所有可能的逆流路径**上。外星人会多次投放兔子。对于每次投放，请找出一个能拦截所有兔子的庄园。若有多个满足条件的庄园，输出其中**编号最大**的一个。

Input

第一行包含两个整数 n, m $(1 \le n \le 2 \times 10^5, n-1 \le m \le 2 \times 10^5)$，分别表示庄园数、水道数。

接下来 m 行，每行包含两个整数 u, v $(1 \le u < v \le n)$，表示一条从 u 流向 v 的水道。输入保证无自环、无重边，且从 1 出发可到达所有节点。

接下来一行包含一个整数 q $(1 \le q \le 2 \times 10^5)$，表示投放的次数。

接下来 q 行，每行描述一次投放：第一个整数 k $(1 \le k \le n)$，随后 k 个互不相同的整数 $x_1, x_2, \cdots, x_k$ $(1 \le x_i \le n)$，表示投放兔子的庄园编号。保证所有询问中 k 的总和不超过 2×10^5。

Output

对于每个询问，输出一行一个整数，即符合条件的庄园编号。

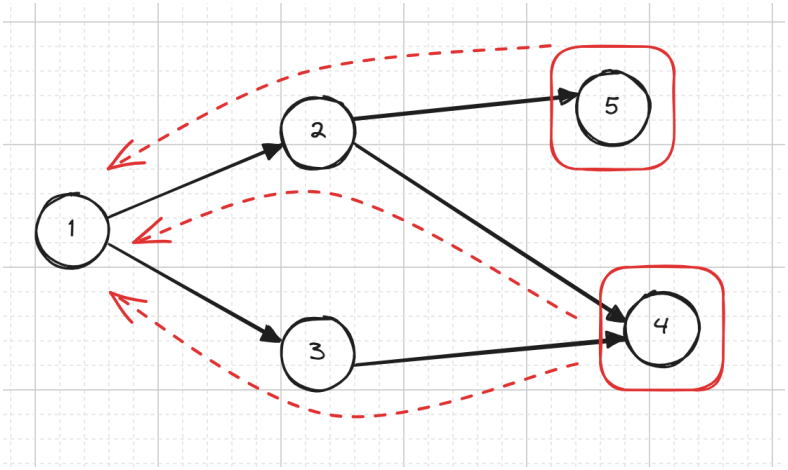
Example

standard input	standard output
5 5 1 2 1 3 2 4 3 4 2 5 3 2 4 5 1 5 2 2 5	1 5 2

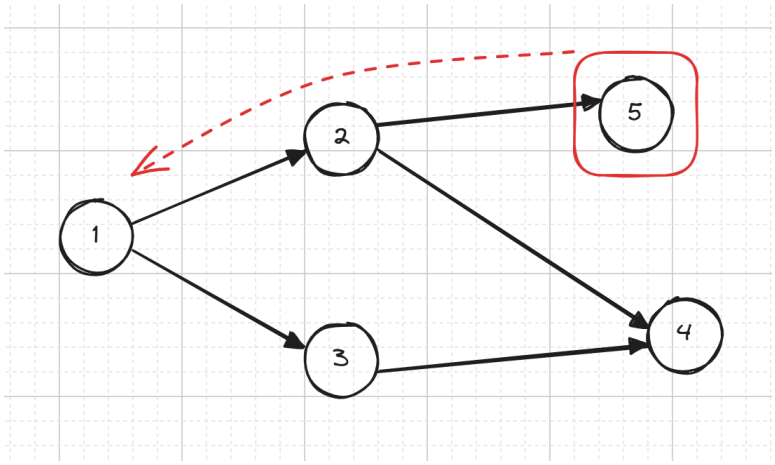
Note

水道构成：$1 \to 2$，$1 \to 3$，$2 \to 4$，$3 \to 4$，$2 \to 5$。

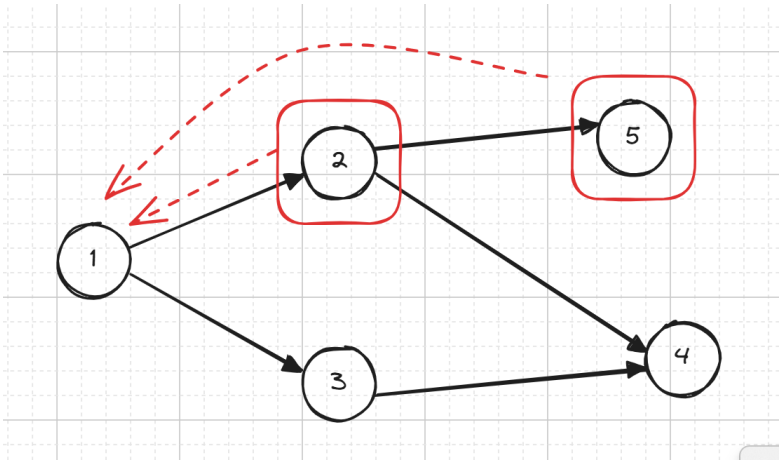
第一个询问：兔子在 $\{4,5\}$，只有在 1 号庄园设防才能拦截所有兔子。



第二个询问：兔子在 $\{5\}$，在 $\{1,2,5\}$ 中任一处设防均可，选择编号最大的 5。



第三个询问：兔子在 $\{2,5\}$，在 $\{1,2\}$ 中任一处设防均可，选择编号最大的 2。



Problem C. 等差数列

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memery limit:	512 megabytes

给定 n 个**互不相同**的整数，请你找出最大的正整数 d，使得存在一个公差为 d 的等差数列（其所有项均为整数），并且这 n 个数都是该等差数列中的项。

Input

第一行包含一个整数 n $(2 \le n \le 10^6)$。

第二行包含 n 个**互不相同**的整数 $A_1, A_2, \cdots, A_n$ $(-10^6 \le A_i \le 10^6)$。

Output

输出一个整数，表示最大的可能公差。

Example

standard input	standard output
3 5 1 3	2
4 9 0 3 15	3

Note

在样例 1 中，公差为 2 的等差数列 $\{\cdots, -1, 1, 3, 5, 7, \cdots\}$ 包含了给定的三个数。

在样例 2 中，公差为 3 的等差数列 $\{\cdots, -3, 0, 3, 6, 9, 12, 15, \cdots\}$ 包含了给定的四个数。

Problem D. 卖货

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memery limit:	512 megabytes

小王有 n 件物品，编号为 $1, 2, \cdots, n$。小吴和小高都想购买这些物品。

对于第 i 件物品，小吴愿意出 b_i 元，小高愿意出 c_i 元。

小王决定从中选择恰好 k 件物品卖给小吴，其余 $n-k$ 件物品全部卖给小高（每件物品只能卖给一个人）。请你计算小王能获得的最大总收入。

Input

第一行包含一个整数 n $(1 \le n \le 10^5)$，表示物品的总数量。

第二行包含 n 个整数 $b_1, b_2, \cdots, b_n$ $(0 \le b_i \le 10^9)$，表示小吴对每件物品的出价。

第三行包含 n 个整数 $c_1, c_2, \cdots, c_n$ $(0 \le c_i \le 10^9)$，表示小高对每件物品的出价。

第四行包含一个整数 k $(1 \le k \le n)$，表示小王打算卖给小吴的物品数量。

Output

输出一行一个整数，表示最大可能的收入。

Example

standard input	standard output
4 9 50 1 4 2 48 5 1 2	66

Note

选择物品 1 和 4 卖给小吴（收入 $9+4=13$），物品 2 和 3 卖给小高（收入 $48+5=53$），总收入 66。没有更高的售卖方式。

Problem E. 和异位

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memery limit:	512 megabytes

给定一个 n 个点 m 条边的无向连通图，每条边有一个非负整数权值。定义一条路径的权值为路径上所有边权的**异或和**（即按位异或运算的结果）。

现有 q 次询问，每次给出两个节点 s 和 t，求从 s 到 t 的所有路径中，权值的最小值。

异或运算：对于两个二进制位，相同得 0，不同得 1。多个数的异或和即为**将它们依次进行异或运算得到的结果**。

Input

第一行包含两个整数 n, m $(1 \le n \le 10^5, n-1 \le m \le 2 \times 10^5)$，分别表示无向图中点和边的数量。

接下来 m 行，每行包含三个整数 u, v, w $(1 \le u, v \le n, 0 \le w < 1 \times 10^{18})$，表示一条连接 u 和 v 的无向边，权值为 w。

保证给定的图联通且无自环，但可能有重边。

接下来一行包含一个整数 q $(1 \le q \le 10^5)$，表示询问的数量。

接下来 q 行，每行包含两个整数 s, t $(1 \le s, t \le n, s \ne t)$，表示一次询问。

Output

对于每次询问，输出一行一个整数，即所求的最小异或值。

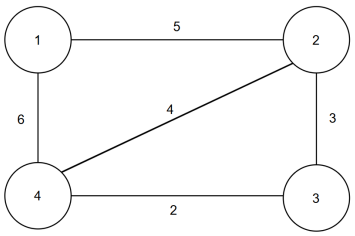
Example

standard input	standard output
4 5 1 2 5 2 3 3 3 4 2 1 4 6 2 4 4 2 1 3 2 4	1 1

standard input	standard output
6 6 1 2 9 2 3 3 3 4 9 3 6 5 4 5 10 5 6 11 2 1 2 2 3	4 3

Note

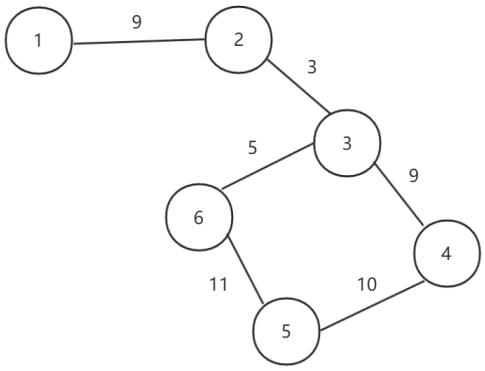
对于样例 1：



询问 $1 \to 3$：路径 $1 \xrightarrow{6} 4 \xrightarrow{4} 2 \xrightarrow{3} 3$ 的异或和为 $6 \oplus 4 \oplus 3 = 1$，不存在更小异或和的路径。

询问 $2 \to 4$：路径 $2 \xrightarrow{3} 3 \xrightarrow{2} 4$ 的异或和为 $3 \oplus 2 = 1$，其他路径均不小于 1。

对于样例 2：



询问 $1 \to 2$：路径 $1 \xrightarrow{9} 2 \xrightarrow{3} 3 \xrightarrow{9} 4 \xrightarrow{10} 5 \xrightarrow{11} 6 \xrightarrow{5} 3 \xrightarrow{3} 2$ 的异或和为：$9 \oplus 3 \oplus 9 \oplus 10 \oplus 11 \oplus 5 \oplus 3 = 4$，不存在更小异或和的路径。

询问 $2 \to 3$：路径 $2 \xrightarrow{3} 3$ 的异或和为 3，不存在更小异或和的路径。

Problem F. 覆盖面积

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 second (8 seconds for Python)
Memery limit:	512 megabytes

给定一个 $n \times m$ 的地图，每个格子有一个整数表示海拔高度。水从水源格子出发，只能流向相邻（上下左右）且海拔**小于等于**当前格子海拔的格子。

每次询问给出一个水源位置 (x, y)，求从该水源出发能够到达的所有格子（包括水源本身）的数量。

Input

第一行包含两个整数 n, m $(1 \le n, m \le 50)$，表示地图的行数和列数。

接下来 n 行，每行包含 m 个整数，表示每个格子的海拔 $A_{i,j}$ $(-10^9 \le A_{i,j} \le 10^9)$。

下一行包含一个整数 q $(1 \le q \le 10^6)$，表示询问次数。

接下来 q 行，每行包含两个整数 x, y，表示水源的坐标（$1 \le x \le n$, $1 \le y \le m$）。

Output

对于每次询问，输出一行一个整数，表示能够覆盖的格子数量。

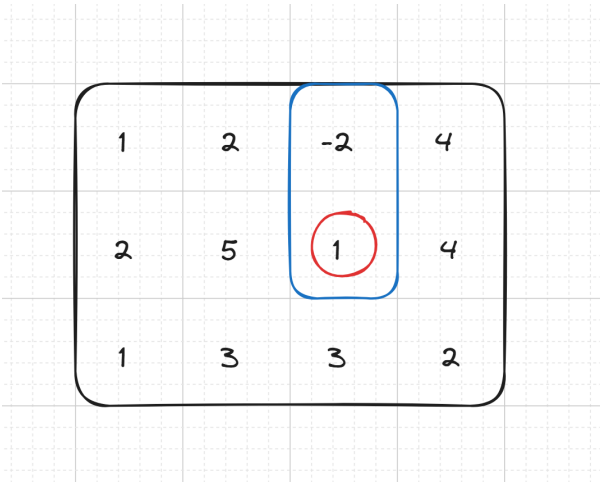
Example

standard input	standard output
3 4 1 2 -2 4 2 5 1 4 1 3 3 2 3 2 3 2 2 2 4	2 10 5

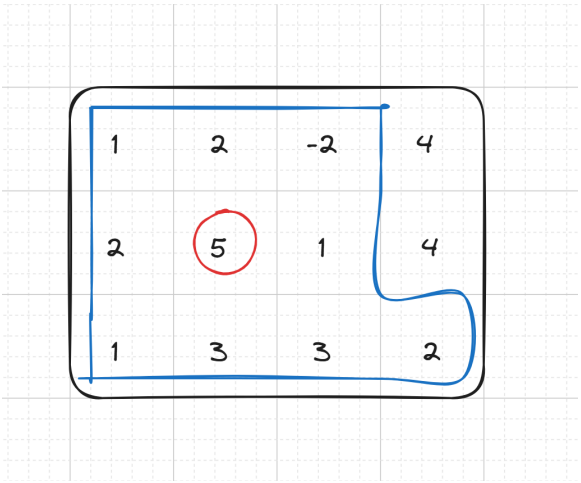
Note

下面用红圈表示水源起点，蓝色的表示被水源覆盖的范围。

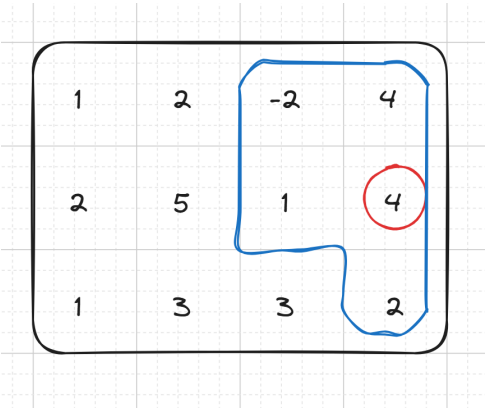
第一个询问：



第二个询问：



第三个询问：



Problem G. 冰淇淋

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memery limit:	64 megabytes

小曾、小王和小刘三人各自有一些零花钱。他们想比比谁的钱最少，决定由钱最少的人请大家吃冰淇淋。请你编写程序，帮他们找出三人中钱数的最小值。

Input

一行包含三个整数 a, b, c ($1 \le a, b, c \le 10^9, a \ne b, a \ne c, b \ne c$)，以空格分隔，分别表示小曾、小王、小刘拥有的钱数（单位：元）。

Output

一个整数，表示三人中钱数的最小值。

Example

standard input	standard output
5 3 8	3
7 7 2	2

Note

对于样例 1，小曾有 5 元，小王有 3 元，小刘有 8 元，钱最少的是小王（3 元）。

对于样例 2，小曾有 7 元，小王有 7 元，小刘有 2 元，钱最少的是小刘（2 元）。

Problem H. 合理分配

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memery limit:	512 megabytes

在 ACM 赛制中，解题数目优先，但有时不得不为题目设置分值。小王负责给一场比赛出题，共有 n 道题目，题目难度**从低到高**依次编号为 $1, 2, \cdots, n$。他需要为每道题设定一个**整数分数** A_i，满足以下条件：

- 分数范围：$1 \le A_i \le m$。
- 难度高的题目分数不低于难度低的题目，即 $A_1 \le A_2 \le \cdots \le A_n$。
- 做题数多的总分一定**严格大于**做题数少的总分。
- 形式化讲：**任意**两组不同的题目集合，若集合大小不同，则较大的集合中所有题目的分数之和**严格大于**较小的集合中所有题目的分数之和。即对于任意两个非空子集 $S, T \subseteq \{1, 2, \cdots, n\}$，若 $|S| > |T|$，则 $\sum_{i \in S} A_i > \sum_{i \in T} A_i$。

小王想知道有多少种不同的分数分配方案满足上述条件。请输出方案数对 998 244 353 取模的结果。

Input

一行两个整数 n, m $(2 \le n, m \le 5\,000)$，分别表示题目总数和分数上限。

Output

输出一个整数，表示方案数对 998 244 353 取模后的结果。

Example

standard input	standard output
2 2	3
3 3	7
5 2	3
100 100	96354127

Note

对于样例 1：所有合法方案为 $(1,1)$、$(1,2)$、$(2,2)$。$(2,1)$ 不合法，因为不满足 $(A_1 \le A_2)$。

对于样例 2：所有合法方案为 $(1,1,1)$、$(1,2,2)$、$(1,3,3)$、$(2,2,2)$、$(2,2,3)$、$(2,3,3)$、$(3,3,3)$。$(1,1,2)$ $(1,2,3)$ 不合法，因为只做前两道题的人和只做最后一道题的人相同得分。$(1,1,3)$ 不合法，因为只做前两题的人总分**小于**只做最后一道题的人。

Problem I. 妙不可言

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds (4 seconds for Java and Python)
Memery limit:	512 megabytes

小龙面前有 n 个盒子，编号 1 到 n。每个盒子中**至多**放有一个马斯卡彭蛋糕，但小龙并不知道这些蛋糕的具体分布。

小刘可以告诉小龙任意一个连续区间 $[l,r]$ 内蛋糕总数的**奇偶性**，但每次询问需要支付一定 $w_{l,r}$ 元。

小龙希望确定哪些盒子内有蛋糕，问**最少**需要支付的总金额是多少。

Input

第一行一个整数 n $(1 \le n \le 2\,000)$，表示盒子的数量。

接下来 n 行，第 i 行有 $n-i+1$ 个整数，其中第 j 个整数表示询问区间 $[i,i+j-1]$ 的价格 $w_{i,i+j-1}$ $(1 \le w_{i,i+j-1} \le 10^9)$。

Output

输出一个整数，表示能够确定所有有蛋糕的盒子最小总金额。

Example 1

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

Note 1

假设：样例 1 的蛋糕分布为：1 0 1 0

第一次询问：$[1,1]$ 花费 1 代价得到区间的奇偶性为 1。

通过第一次询问可以推出，1 位置有蛋糕。

第二次询问：$[1,2]$ 花费 2 代价得到区间的奇偶性为 1。

通过第一次和第二次询问可以推出，2 位置没有蛋糕。

第三次询问：$[3,3]$ 花费 3 代价得到区间的奇偶性为 1。

通过第三次询问可以推出，3 位置有蛋糕。

第四次询问：$[3,4]$ 花费 1 代价得到区间的奇偶性为 1。

通过第三次和第四次询问可以推出，4 位置没有蛋糕。

可以证明，无论蛋糕如何分布，通过上述四次询问，都可以唯一确定每个盒子内的状态。且不存在更小的花费方案能够唯一确定蛋糕状态。

Example 2

standard input	standard output
4 1 2 2 6 4 2 3 3 3 4	8

Note 2

一种可行的方案如下：

第一次询问：$[1,1]$ 花费 1 代价得到区间的奇偶性。

通过第一次询问可以推出，1 位置的状态。

第二次询问：$[1,3]$ 花费 2 代价得到区间的奇偶性。

第三次询问：$[2,4]$ 花费 3 代价得到区间的奇偶性。

第四次询问：$[1,2]$ 花费 2 代价得到区间的奇偶性。

通过 1 位置的状态和第四次询问 $[1,2]$ 可以推出，2 位置的状态。

通过 $1,2$ 位置的状态和第二次询问 $[1,3]$ 可以推出，3 位置的状态。

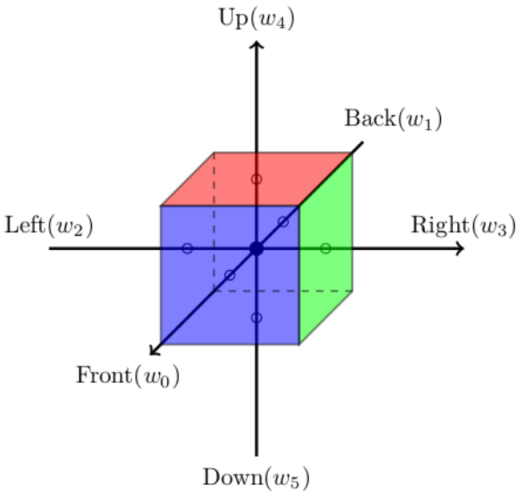
通过 $2,3$ 位置的状态和第三次询问 $[2,4]$ 可以推出，4 位置的状态。

不存在更小的花费方案。

Problem J. 骰子

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memery limit:	64 megabytes

我们有一个六面骰，每个面都有一个数字。骰子的六个面按照 **前**、**后**、**左**、**右**、**上**、**下** 的顺序给出初始数字，如下图所示：



初始时，骰子各面的数字分别为 $w_0, w_1, w_2, w_3, w_4, w_5$，分别对应骰子的六个面。

随后，骰子将进行 n 次滚动，每次滚动方向为 `F`，`B`，`L`，`R` 之一，分别表示：

- `F`：骰子向正对我们的方向滚动（即向前滚动）。
- `B`：骰子向背对我们的方向滚动（即向后滚动）。
- `L`：骰子向左滚动。
- `R`：骰子向右滚动。

你需要输出经过 n 次滚动后骰子底面的数字。

Input

第一行包含六个整数 $w_0, w_1, w_2, w_3, w_4, w_5$ $(0 \le w_i \le 10^9)$，分别表示初始时骰子前、后、左、右、上、下各面的数字。

第二行包含一个整数 n $(1 \le n \le 10^5)$，表示滚动的次数。

第三行包含一个长度为 n 的字符串 S $(S_i \in \{\texttt{F}, \texttt{B}, \texttt{L}, \texttt{R}\})$，表示每次滚动的方向。

Output

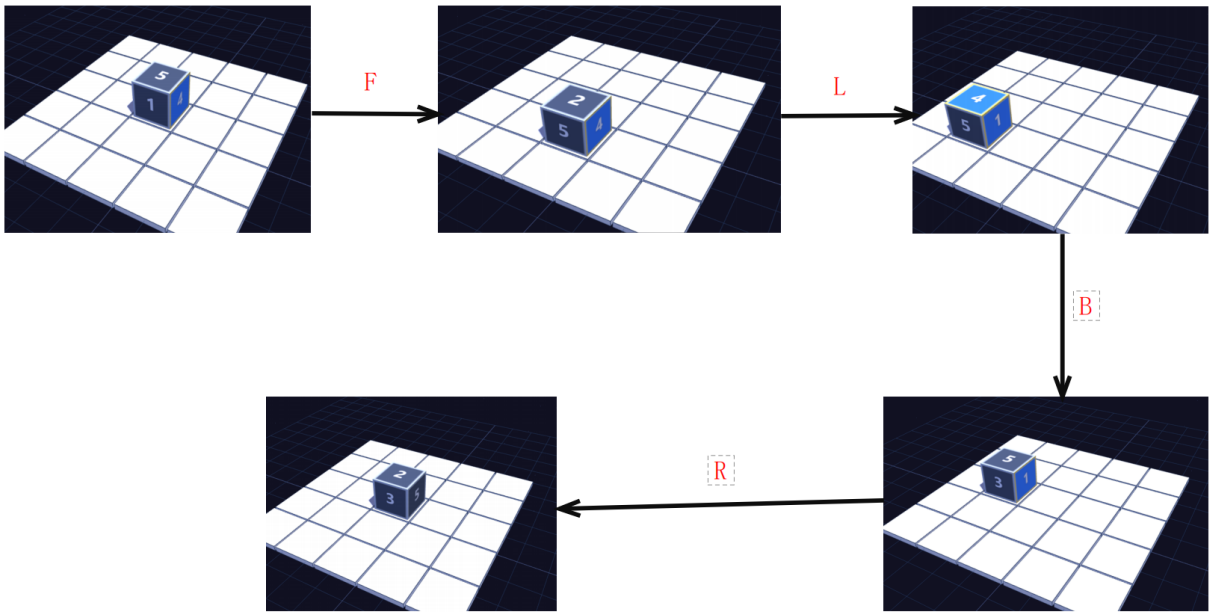
输出一行一个整数，表示滚动结束后骰子底面的数字。

Example

standard input	standard output
1 2 3 4 5 6 4 FLBR	1

Note

样例中的初始状态和每步状态如下：



Problem K. 巴啦啦能量

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memery limit:	64 megabytes

在魔法世界里，小曾需要收集“偶数能量”。能量存在于自然数中，只有偶数才能被提取。这一天，魔法师给了他一个新任务：在区间 $[n, m]$ 内（包含端点），找出所有偶数，并计算它们的能量总和。

请你编写程序帮助小可快速计算出结果。

Input

一行，两个整数 n 和 m $(1 \le n \le m \le 10^8)$，以空格分隔。

Output

一个整数，表示 n 到 m 之间所有偶数的和。

Example

standard input	standard output
2 5	6
3 9	18

Note

对于样例 1，区间 $[2, 5]$ 内的偶数为 $2, 4$，和为 6。

对于样例 2，区间 $[3, 9]$ 内的偶数为 $4, 6, 8$，和为 18。

Problem L. 贪吃的猪

Input file:	standard input
Output file:	standard output
Time limit:	1 second (2 seconds for Java)
Memery limit:	512 megabytes

小高开了一家餐厅，每天厨师会列出 n 道菜，第 i 道菜的售价为 A_i。

小高希望客人吃得更优惠，因此他决定删除一段**连续的菜品**（即一个**连续子数组**），但删除后必须**至少保留第一道菜和最后一道菜**（不能删除头尾菜品）。

删除后，剩下的菜品按原顺序排列，小高希望这些剩余菜品的**平均售价**尽可能低。

请你计算这个最低的平均售价。

Input

第一行一个整数 n ($2 \le n \le 10^5$)，表示菜品的数量。

第二行 n 个整数 $A_1, A_2, \cdots, A_n$ ($1 \le A_i \le 10^9$)，表示每道菜的售价。

Output

一行一个实数，表示最低平均售价。

当且仅当你的答案和正确答案的绝对误差或相对误差不超过 10^{-4} 时，你的答案会被视为正确。即假设你的答案为 a，正确答案为 b，当且仅当 $\frac{|a-b|}{\max(1,|b|)} \le 10^{-4}$ 时，你的答案会被视为正确答案。

Example

standard input	standard output
6 100 1 100 100 2 100	50.75

Note

删除区间 $[3,4]$ 之后剩余四道菜，售价分别为 $100, 1, 2, 100$，此时平均售价为 50.75，不存在比 50.75 更小的平均售价。