

Problem A. 数列

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

某数列的通项公式为 $a_i^2 = S_{i-1}^2 + S_{i-1} + 1$, ($i \geq 2$)。已知 $a_1 = 1$, a_i 为正数。
为了保证结果是正整数, 对于每个给定的 n , 请你输出 $a_n \cdot 2\sqrt{3} \cdot \sin(\frac{\pi}{3 \cdot 2^{n-1}})$ 的值

Input

第一行包含一个整数 n , ($1 \leq n \leq 1000000$)。

Output

请输出 $a_n \cdot 2\sqrt{3} \cdot \sin(\frac{\pi}{3 \cdot 2^{n-1}})$, 每个输出占一行。

Example

standard input	standard output
1	3

Problem B. 期末复习

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

《计算机组成原理》是一门十分重要的课，因而 XZW 并不希望挂科。没办法，只能通宵了。

课本有 N 页，每页的知识点数量为 a_i ，为了尽量将所有知识点都看到，XZW 只会看知识点数量不超过 X 的页面，但这样会留下一些章节没有看到，为了便于之后的复习，XZW 会将所有没看到的页面编号，连续未看的页面会被视为一部分，XZW 希望知道对于给定的每个 X ，没看到的页面有几部分。

为了更好地设置复习计划，XZW 会询问 Q 次，每次会给出一个 X ，请你回答没看到的页面被分成了几部分。

Input

第一行输入一个整数 $N(1 \leq N \leq 200000)$ 代表页数。

第二行有 N 个整数，对于每个整数 $a_i(1 \leq a_i \leq 10^9)$

第三行有一个整数 $Q(1 \leq Q \leq 200000)$ 代表询问次数。

第四行有 Q 个整数，每个整数代表 XZW 每页看的最大的知识量不超过 $X(1 \leq X \leq 10^9)$

Output

对于每次询问，输出一个整数代表答案，每个答案占一行。

Example

standard input	standard output
3	0
1 2 3	1
3	1
3 2 1	

Note

每次询问是独立的。

Problem C. 基因退化

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

人类的基因是一个由小写字母构成的字符串 S_1 。由于近年来人们好吃懒作，他们的基因开始退化了！慌张的科学家们发现他们不能阻止基因的退化，但他们发现了一些规律：基因中的一个已知的子序列 S_2 会从基因当中消失，之后人类的剩下的基因就会散成好几段，而这几段中最长的一段会成为新的基因（其他较短的片段会被淘汰）。

幸运的是，我们现在知道了会消失的子序列是什么。现在想问，新基因可能具有的最长长度是多少。

Input

第一行包含一个字符串， S_1 代表退化之前的基因。

第二行包含一个字符串， S_2 代表消失的子序列。

(S_1 、 S_2 的长度小于等于 100000)

Output

输出新基因可能具有的最长长度。

Example

standard input	standard output
bbaba bb	3

Note

样例解释：bbaba 有 3 种退化情况：

1. 第一个 b 和第二个 b 消失剩下 aba
2. 第一个 b 和第三个 b 消失剩下 ba 和 ba
3. 第二个 b 和第三个 b 消失剩下 b、a 和 a

因此新基因最长因该是 aba 长度为 3

Problem D. 秦神的签到题 2

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

秦神手上有一个长度为 n 序列 \mathbf{a} ，他定义函数 $f(l, r)$ 表示序列 \mathbf{a} 中连续子序列 a_l, a_{l+1}, \dots, a_r 中不同数的数值总和。如 $[2, 1, 2, 1, 3]$ 中不同的数有 $[1, 2, 3]$ ，其总和为 $1 + 2 + 3 = 6$ 。

秦神在这之前已经丢给过你一个简单的问题，让你求所有满足 $1 \leq l \leq r \leq n$ 的 $f(l, r)$ 的和，但是现在他想问你一个更难的问题。

定义一个存储不重复元素的容器为分类器。在分类器中的每一个元素有两个属性，分别是键值和映射值。每一个元素的**键值**是 \mathbf{a} 中的一个连续子序列，每一个元素的**映射值**是这个子序列中不同数的数值总和。分类器只会存储不同的键值，因此重复的额外元素（具有相同的键值）会被移除。

记 $S(l, r)$ 为 a_l, a_{l+1}, \dots, a_r ，即 \mathbf{a} 的一个连续子序列。现在我们用一个分类器 \mathbf{CA} 存储所有满足 $1 \leq l \leq r \leq n$ 的 \mathbf{a} 的连续子序列 $S(l, r)$ 和 $f(l, r)$ 。你需要计算出 \mathbf{CA} 中**所有映射值的和**。

实际上，上述分类器的定义在 C++ 中就是 `map<vector<int>, int>`，在 Java 中就是 `Map<ArrayList<Integer>, Integer>`。如果你熟悉这些数据结构，你要做的就是把所有可能的 $(S(l, r), f(l, r))$ ($1 \leq l \leq r \leq n$) 插入分类器 \mathbf{CA} ，并计算其中所有映射值的和。由于答案可能很大，你只需要输出答案对 998244353 取模的结果。

Input

第一行有一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示序列 \mathbf{a} 的长度。

第二行有 n 个整数 a_1, a_2, \dots, a_n 表示序列 \mathbf{a} ($a \leq a_i \leq 10^9$)。

Output

输出一个整数，表示分类器 \mathbf{CA} 所有映射值的和对 998244353 取模后的结果。

Examples

standard input	standard output
5 1 2 3 4 5	105
3 2 3 3	18

Problem E. 110!!

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

有一个很大的正整数 N ，现在从这个整数中截取连续的一部分得到整数 S ，要求整数 S 能被 110 整除 (不计算 S 等于 0 的情况)，问这样不同的 S 有多少个。 S 在 N 中的位置不同即可认为是不同的。

Input

输入仅一行，一个正整数 N ， $1 \leq N \leq 10^{5000}$ 。

Output

输出一个整数代表 S 的个数，每个输出占一行。

Example

standard input	standard output
110110	3

Note

样例中的 S 可以是 110、110、110110

Problem F. 长方体

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

有一个长方体, 长、宽、高的长度为 L 、 W 、 H , 将这个长方体放到一个三维直角坐标系中, 选择其底面的一个顶点作为坐标原点, 现在要从 $(0, 0, 0)$ 点到达 (L, W, H) 点, 每次只能沿一个方向走一个单位长度且只能向坐标递增的方向走。对于 (X, Y, Z) 只能走到 $(X+1, Y, Z)$ 或者 $(X, Y+1, Z)$ 或者 $(X, Y, Z+1)$ 。在这个坐标系中, 有两个点是不可达的, 这两点的坐标为 (X_1, Y_1, Z_1) , (X_2, Y_2, Z_2) , 问 $(0, 0, 0)$ 到达 (L, W, H) 的方案数。

Input

第一行三个整数 L, H, W ($1 \leq L, H, W \leq 10^5$) 第二行有六个整数 $X_1, Y_1, Z_1, X_2, Y_2, Z_2$, ($1 \leq X_1, X_2 \leq L, 1 \leq Y_1, Y_2 \leq H, 1 \leq Z_1, Z_2 \leq W$)

Output

仅一行代表方案数。结果可能很大, 请对 1000000007 取余。

Example

standard input	standard output
1 1 1 1 0 1 0 1 1	2

Problem G. 自走棋!

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

自走棋已经成为了当今最风靡的休闲竞技类游戏! 它的核心玩法是选取一些棋子, 让他们自己战斗! 不仅棋子本身有自己的战斗力, 一些同种族的棋子同时在场时, 你还会获得种族加成!

自走棋的棋子和种族还是比较多样的! 对于一款新的自走棋游戏, 玩家们在刚玩的时候都会想尽快知道, 怎么才能组成最强阵容!

假设现在有你可以上场 n 个棋子, 棋子有 m 种种族! 对第 i 个种族, 总共有 c_i 个该种族的棋子, 每个棋子有对应的单独战力 p ! 当满足加成规则的时候, 可以获得种族 buff, 你的战斗力又会额外加 add_i ! (为了使问题更加简单, 你只有让这些棋子全部上场的时候, 才能获得这个种族的战力加成!)

现在问你, 你的阵容能拥有的最大战力是多少!!!

Input

第一行是两个正整数 n, m , 分别代表可以上场的棋子数和以知的种族数 ($m \leq 10$)

第二行有 m 个正整数 c_i , 代表对应第 i 个种族有 c_i 个棋子。($\sum_{i=1}^m c_i \leq 1000$)

第三行又有 m 个正整数 add_i , 代表对应第 i 个种族的棋子凑齐时能获得 add_i 的战力加成。

接下来 m 行分别有 c_i 个正整数 p , 代表第 i 个种族的每个棋子的单独战力!

Output

输出阵容能获得的最大战力 (保证答案能用长整形表示)!

Example

standard input	standard output
4 2 3 2 0 10000 1000 1000 1000 1 1	12002

Note

样例解释: 现在有 2 个种族, 第一个种族有 3 个棋子, 第二个种族有 2 个棋子。当你凑满第一个种族的棋子时, 获得的战力加成为 0。当你凑满第二个种族的棋子时, 获得的战力加成为 10000。现在要取 4 个棋子, 最优的解是把两个战力为 1 的二号种族棋子选上, 这样能获得 10000 点的战力加成, 接下来再选两个战力为 1000 的一号种族棋子

Problem H. 学车

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

ZZY 在报名学车两年以后终于开始学车了，因为这是最后一年的期限了。他的驾校教练在看到他以后，把他臭骂了一顿。在他听到 ZZY 是上海大学 ACM 队的队员时，他决定考考 ZZY，于是给他出了一道题。ZZY 觉得太简单了，于是把题目留给今天的小朋友们。

问题是这样的：ZZY 的教练暑假的时候有很多学员约他学车，总共有 n 个学员，他们分别有自己的学车时间 $S_i \sim E_i$ 学员的练车时间可能有重叠（注意： E_i 和 S_i 代表的是某个时刻，如果在某时刻有一位学员结束练车另外一位学员开始练车，那么这两位学员的练车时间是不重叠的）。因为，为了让所有人都能学到车，所以教练车上就可能同时有几个学员。教练担心自己的车子可能会超载，所以他想提前知道自己得开一辆载客量至少是多少的车子来驾校。

Input

第一行是一个整数 n ，代表学员的总数 ($1 \leq n \leq 500000$) 接下来 n 行，每行输入两个整数 S_i 和 E_i ，代表第 i 位学员学车开始的时间和结束的时间 ($1 \leq S_i < E_i \leq 1000000000$)

Output

输出一个整数，代表教练车的至少载客数（这个数目可能很大，但是 ZZY 的教练什么车都有，所以不用担心）

Examples

standard input	standard output
3 1 3 2 4 5 6	2
5 3 5 5 6 1 3 1 5 10 100	2