

T1-中彩票

题目描述

今天是个好日子，小明和小红想买个彩票试一下自己的手气。他们买的彩票是这样兑奖的，只要买到的彩票号码中包含长度大于等于 2 的回文数就判定中奖了。现在请你编写一个程序帮小明和小红检查一下他们是否中奖。

提示：回文数指的是从左向右读和从右向左读都是一样的整数。例如 22 是长度为 2 的回文数，654456 是长度为 6 的回文数。

输入格式

共三行。

第一行为一个整数 n ，代表小明和小红买到的彩票号码长度。

第二行为小明买到的彩票号码。

第三行为小红买到的彩票号码。

输出格式

只有一个单词。若两人均中奖，输出 “All”；若小明中奖，输出 “Ming”；若小红中奖，输出 “Hong”；若两人均未中奖，则输出 “No”。

样例输入

6

172275

654321

样例输出

Ming

样例解释

小明的彩票号码中有 7227 这个长度为 4 的回文数，而小红的彩票号码中没有出现回文数，所以只有小明中奖了。

数据范围

对于 60% 的数据，保证输入的 $n \leq 10000$ ，且彩票号码符合特殊性质 1。

对于 100% 的数据，保证输入的 $n \leq 1000000$ ，数据点 7-8 符合特殊性质 2，数据的 9-10 无特殊性质。

特殊性质解释

特殊性质 1：彩票号码只有两种情况，要么自己本身是一个回文数，要么不包含回文数。

特殊性质 2：彩票号码若包含回文数，则回文数一定在正中间。

T2-数字游戏

题目描述

小明和小红在做一个数字游戏。小明写出一个二进制数，小红必须交换其中的两个数字来得出一个新的二进制数（必须进行交换操作且只能交换两个数字）。小明想知道所有得到的新二进制数中第二大的数是几。小红知道怎么计算最大的数，但是她不知道怎么计算第二大的数，你能帮帮她么？

输入格式

共一行。为小明给出的一个二进制数，不包含前导零。保证所有输入的二进制数通过交换一定能产生至少两个比原数更大的数字。

输出格式

共一行。为小红执行交换操作后第二大的二进制数。

样例输入

1110011

样例输出

1111001

样例解释

交换后最大的二进制数为 **1111010**，交换的是第四位的 **0** 和第 7 位的 **1**。第二大的二进制数为 **1111001**，交换的是第四位的 **0** 和第 6 位的 **1**。

输入解释

如数字:**11110000**，不论如何交换，交换后一定会变小，这种输入情况不会存在；再比如数字：**101000**，交换后只会产生一个比他大的数字 **110000**，无法通过交换得到第二个比原数字更大的数，这种输入情况也不会存在。

数据范围

对于 30%的数据，保证输入的二进制数长度 ≤ 10 。

对于 60%的数据，保证输入的二进制数长度 ≤ 1000 。

对于 100%的数据，保证输入的二进制数长度 ≤ 1000000 。

T3-加密通信

题目描述

在一次军事演习中，小凯担任通信兵，负责解密传来的信息。他会收到一份 $n*n$ 的数字表（仅包含 0-9 共 10 种数码），和一份 $m*m$ 的字母表（仅包含 A-Z、a-z 共 52 种字符）。

数字表被称为解密卡，字母表被称为加密卡。规定如下名词：

1.覆盖区域：将数字表和字母表对齐，使得数字表在上，字母表在下，数字与字母一一对应，如果数字表的 1 行 1 列对应字母表的 x 行 y 列，那么数字表的 $1+k$ 行 $1+k$ 列对应字母表的 $x+k$ 行 $y+k$ 列，此时覆盖区域为字母表以 x 行 y 列为左上角、 $x+n-1$ 行 $y+n-1$ 列为右下角的正方形范围。简单来说，根据图片显示，若数字表的左上角盖在了 2 行 2 列的位置，则数字表会覆盖（2，2）到（3，3）这片区域。

2.有效区域：称某字母和其在字母表的顺序数字为互相对应，字母 A 和 a 是第 1 个字母、字母 I 和 i 是第 9 个字母，那么 A、a 和 1 对应，I、i 和 9 对应，以此类推。如果覆盖区域的四个顶点字母恰好和原始数字表的四个顶点数字互相对应，那么这个覆盖区域称之为有效区域。

数字表		字母表			
1	2	z	z	z	z
3	4	z	a	B	z
		z	c	D	z
		z	z	z	z

3.数字关联：0 和任何字母关联，1 不能和任何字母关联，2-9 中的质数和大写字母关联，2-9 中的合数和小写字母关联。若关联成功，则提取出来作为密码的一部分，如上图中 2 是质数，需关联大写字母，2 对应的是 B，关联成功；3 是质数，需关联大写字母，3 对应的 c 关联失败；4 是合数，需关联小写字母，而 4 对应的是 D，关联失败。因此上述图中，本次覆盖只关联成功一次，提取出一个密码 B。

4. 解密数字表：大小为 $n*n$ ，可以由原始数字表顺时针旋转若干次 90° 得到。

1	2	顺时针 旋转90°	3	1
3	4		4	2

5.区域解密：有效区域和解密数字表对齐，从左到右从上到下，将和解密数字表中对应位置数字恰好关联的字母提取出来，构成字符串。

一次解密操作包含以下步骤：首先从左到右从上到下找出所有有效区域，依次进行解密。对于每个有效区域首先用原始数字表进行 1 次区域解密，然后根据当前有效区域的顶点大写字母数量 x ，额外进行 x 次区域解密（如图示中四个顶点有两个大写字母，因此需旋转两次），第 i 次解密时将原始数字表顺时针旋转 $i*90^\circ$ 后作为解密数字表再进行区域解密。最后将所有区域解密的字符串按序连接，构成最终的解密字符串。

你需要帮助小凯完成解密工作。

输入格式

共 $n+m+1$ 行，

第 1 行两个正整数 n 和 m ，分别代表数字表和字母表的大小；

第 2 至 $n+1$ 行分别有连续的 n 个数码字符，代表数字表的内容；

第 $n+2$ 至 $n+m+1$ 行分别有连续的 m 个字母字符，代表字母表的内容。

输出格式

仅一行一个字符串，代表解密后的答案。如果答案为空串，你需要输出 “No solution”。

样例输入 1

2 3

11

13

AAB

ACB

BzB

样例输出 1

CAAAC

样例输入 2

3 4

101

245

313

DABa

AFab

Fcdc

cdcD

样例输出 1

BFaABabFFcAFcd

数据范围

对于 30% 的数据，保证输入的 $m \leq 10$ 。

在前 30% 中有 10% 的数据，保证数字表中仅含有 0 和 1。

对于 70% 的数据，保证输入的 $m \leq 50$ 。

在前 70% 中有 30% 的数据，保证字母表中仅含有小写字母。

对于 100% 的数据，保证输入的 $n \leq m \leq 80$ 。

样例解释 1

仅有 (1,1) (2,2) 这一个有效区域，此区域的四个顶点共有 4 个大写字母需要额外进行 4 次区域解密，此区域第 1 次解密为 C（仅在 (2,2) 位置有质数 3 与大写字母 C 关联），第 2-4 次为 A，第 5 次为 C，最终答案为 CAAAC。

样例解释 2

有 (1,2) (3,4) 以及 (2,1) (4,3) 这两个有效区域，第一个 (1,2) (3,4) 区域为

ABa

Fab

Cdc

此区域的四个顶点共有 1 个大写字母需要额外进行 1 次区域解密。此区域的第一次解密得到的密码为 BFa，旋转一次密码卡，进行第二次解密，得到的密码为 ABab；第二个区域为

AFa

Fcd

cdc

此区域的四个顶点共有 1 个大写字母需要额外进行 1 次区域解密。此区域的第一次解密得到的密码为 FFc，旋转一次密码卡，进行第二次解密，得到的密码为 AFcd；因此总密码为四块密码从前往后拼在一起：BFaABabFFcAFcd。

T4-星际穿梭

题目描述

为了快速在宇宙中穿梭，人类研发了一款能够瞬间加速的宇宙飞船。

这款飞船起步时可以任意选择速度 a 千米每秒， a 必须为正整数。

另外此飞船的发动机有一个长度为 m 的加速参数序列 b_1 、 b_2 、...、 b_m ，起步后第 i 秒可以根据发动机参数 b_i 来调整飞船速度，使其至少增加 2 倍、至多增加 b_i 倍；如 $b_i=3$ ，则可以选择加速 2 倍或 3 倍。

特别的，若 i 大于 m ，那么加速参考 b_m ，即速度至少增加 2 倍、至多增加 b_m 倍。如加速参数只有两条 b_1 、 b_2 ，则第三秒时加速参考 b_2 。增加的速度必须为原本速度的整数倍。飞船改变速度后的 1 秒内，飞船都会按此速度飞行。

你是其中一艘宇宙飞船的驾驶员，现在你需要执行飞行任务，经过 n 个排成一条直线的空间站，第 i 个空间站在起点前方距离 c_i 千米。如果飞船正好在整数秒抵达空间站，那么这个空间站的信息可以被飞船获得。你需要计算在获得所有空间站信息的基础上，飞行时间如何尽可能的短，输出这个最短飞行时间秒数。如果不能获取所有空间站信息，请输出 -1。

输入格式

共四行

第一行一个整数 m ，代表飞船的发动机参数序列长度；

第二行 m 个整数 b_1 、 b_2 、...、 b_m ，代表飞船的发动机参数序列；

第三行一个整数 n ，代表空间站数量；

第四行 n 个整数 c_1 、 c_2 、...、 c_n ，代表起点正前方空间站的距离。

输出格式

仅一个正整数，代表最短飞行时间秒数或者 -1。

样例输入 1

```
2
9 9
1
571
```

样例输出 1

```
5
```

数据范围

对于 30% 的数据，保证输入的 $\max\{c_1, c_2, \dots, c_n\} \leq 30$ ；其中数据 1， $m=n=1$ ；

对于 70% 的数据，保证输入的 $\max\{c_1, c_2, \dots, c_n\} \leq 100000$ ；

对于 100% 的数据，保证输入的 $\max\{c_1, c_2, \dots, c_n\} \leq 10^9$ ， $n, m, \max\{b_1, b_2, \dots, b_n\} \leq 20$ 。

样例解释 1

初始速度为 1km/s, 0-1s 走了 1km；1s 时将速度调整 2 倍为 2km/s，1-2s 走了 2km；2s 时将速度调整 4 倍为 8km/s，2-3s 走了 8km；3s 时将速度调整 7 倍为 56km/s，3-4s 走了 56km；4s 时将速度调整 9 倍为 504km/s，4-5s 走了 504km。5s 一共走了 $1+2+8+56+504=571$ km。