Problem A. Yukina with n-Power

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

请输出$(3+\sqrt{5})^n$整数部分最后3位。如果结果不超过2位，请补足前导0。

Input

输入一个正整数$n(2 <= n <= 2000000000)$

Output

输出计算结果

Samples

standard input	standard output
18	607
2	027

Problem B. Yukina with Three Points

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

在$x-y$坐标系里面有n个分布不同坐标位置的点(x_i, y_i)（已知n个点都不重合）。

现在需要在这n个点里面找到有多少由三个点A, B, C组成，且满足点B是点A和点C中点的组合数。

这里假设组合$(A-B-C)$和组合$(C-B-A)$是同一组。

Input

第一行输入正整数$n(3 <= n <= 3000)$，表示坐标轴内点的对数

接下来n行，每行两个正整数x和$y(-1000 <= x, y <= 1000)$，表示某个点的坐标

Output

输出满足条件的点的组合数

Sample

standard input	standard output

4
0 0
1 0
2 0
3 0
	2

Problem C. Yukina with Courses

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

Yukina has N courses this term, and she plans to spend at most M days on study.

Of course, the profit she will gain from different course depending on the days he spend on it.

How to arrange the M days for the N courses to maximize the profit?

Input

The input consists of multiple data sets.

A data set starts with a line containing two positive integers N and M, N is the number of courses, M is the days Yukina has.

Next follow a matrix $A[i][j]$, $(1 <= i <= N <= 100, 1 <= j <= M <= 100)$.$A[i][j]$ indicates if Yukina spend j days on i^{th} course she will get profit of value $A[i][j]$.$N = 0$ and $M = 0$ ends the input.

Output

For each data set, your program should output a line which contains the number of the max profit Yukina will gain.

Sample

standard input	standard output

2 2
1 2
1 3
2 2
2 1
2 1
2 3
3 2 1
3 2 1
0 0
	3
	4
	6

Problem D. Yukina with NEWS

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

给定一串只包含$'N','E','W','S'$的字符串代表一条路径，你可以选定其中连续的一段将其进行路径压缩，至多压缩一次

具体操作为：对于路径串$R = r_1, r_2, \cdots, r_n$，挑选合适的$i$与$j(1 \le i \le j \le n)$，记$P = r_i, r_{i+1}, \cdots, r_j$，可以将$R$中所有不重复的$P$替换为$'*'$，即可使用$P$串与压缩后的$R$串表示原路径

例如，对于$R = ''NWEWWSWEWWEWWNEWSWEWW''$，若取$P = ''WEWW''$，可将$R$压缩为$''N*S*EWWNEWS*''$或$''N*SWEW*NEWS*''$，即可使用4+12=16个字符表示原先21个字符的串

Input

输入包含且仅包含一行仅包含$'N','E','W','S'$的方向字母串$R(1 \le |R| \le 10^2)$

Output

输出要表示该串的最少总字母数

Sample

standard input	standard output
NWEWWSWEWWEWWNEWSWEWW	16

Problem E. Yukina with Virginia

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

Yukina学会了一种RHY密码，明文与密文对应表如下

明文	A	B	C	D	E	F	G	H	I	J	K	L	M
密文	1	2	3	4	5	6	7	8	9	10	11	12	13

明文	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
密文	14	15	16	17	18	19	20	21	22	23	24	25	26

对大写字母加密后，变为一连串的数字，但是加密后的密文连在一起了

例如RHY加密后变为18825，可以被解读为以下5种

- AHHBE=1/8/8/2/5
- RHBE=18/8/2/5
- RHY=18/8/25
- AHHY=1/8/8/25

Input

有且仅有一行$S(1 \leq |S| \leq 10^5)$

保证S是由一个合法的大写字母串加密而来

Output

输出可能的解读总数，对1000000007取模

Samples

standard input	standard output
1234	3
22027	1

Note

对于样例1，可以解释为1/2/3/4，12/3/4，1/23/4

对于样例2，只能解释为2/20/2/7

Problem F. Yukina with LST

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

Yunika被LST迫害在了一个10×10的荒野里

需要找到位于‘T’的出口才能逃离，LST也在这片荒野里，但他不会动!

- ’L’ - LST
- ‘S’ - Yukina的初始位置
- ’T’ - Yukina的目标位置

Input

一个10行10列的矩阵，见样例，保证地图中包含恰好一个‘L’，一个‘S’和一个‘T’

Output

输出从‘S’走到‘T’所需的最少步数

Sample

standard input	standard output
....S....... ....L.... ....T....	8

Problem G. Yukina with a Simple Problem On LIS

Input file:	`standard input`	Time limit:	3 seconds
Output file:	`standard output`	Memory limit:	256 megabytes

给定一个长度为 n 的数列 a ，我们等概率地取出任意一个上升子序列，求每个数被选中的概率，对 998244353 取模。

Input

第一行输入一个整数 $n(1 \leq n \leq 5 \times 10^5)$ 表示数列 a 的长度。

接下来一行输入 n 个整数 $a_i(1 \leq a_i \leq 10^9)$🄵。

Output

输出 n 个整数，表示每个数字被选中的概率。

Sample

standard input	standard output

`4`
`1 2 2 4`
	`635246407 90749487 90749487 635246407`

Note

模意义下的除法 $\frac{a}{b}$ ，可以写成 $ab^{-1}(modP)$ ，也就是 a 乘以 b 的逆元。

Problem H. Yukina with Knights

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

Berland is facing dark times again. The army of evil lord Van de Mart is going to conquer the whole kingdom. To the council of war called by the Berland's king Valery the Severe came n knights. After long discussions it became clear that the kingdom has exactly n control points (if the enemy conquers at least one of these points, the war is lost) and each knight will occupy one of these points.

Berland is divided into $m + 1$ regions with m fences, and the only way to get from one region to another is to climb over the fence. Each fence is a circle on a plane, no two fences have common points, and no control point is on the fence. You are given k pairs of numbers a_i, b_i . For each pair you have to find out: how many fences a knight from control point with index a_i has to climb over to reach control point b_i (in case when Van de Mart attacks control point b_i first). As each knight rides a horse (it is very difficult to throw a horse over a fence), you are to find out for each pair the minimum amount of fences to climb over.

Input

The first input line contains three integers $n, m, k (1 \le n, m \le 1000, 0 \le k \le 100000)$. Then follow n lines, each containing two integers $K_{x_i}, K_{y_i} (-10^9 \le K_{x_i}, K_{y_i} \le 10^9)$ — coordinates of control point with index i. Control points can coincide.

Each of the following m lines describes fence with index i with three integers $r_i, C_{x_i}, C_{y_i} (1 \le r_i \le 10^9, -10^9 \le C_{x_i}, C_{y_i} \le 10^9)$ — radius and center of the circle where the corresponding fence is situated.

Then follow k pairs of integers $a_i, b_i (1 \le a_i, b_i \le n)$, each in a separate line — requests that you have to answer. a_i and b_i can coincide.

Output

Output exactly k lines, each containing one integer — the answer to the corresponding request.

Sample

standard input	standard output

`2 1 1`
`0 0`
`3 3`
`2 0 0`
`1 2`
	`1`

Problem I. Yukina with Square

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

在一个 $n \times m$ 的只包含 0 和 1 的矩阵里找出一个不包含 0 的最大正方形，输出边长。

Input

第一行为两个整数 $n, m (1 \le n, m \le 10^3)$，接下来 n 行，每行 m 个数字，用空格隔开，0 或 1 。

Output

一个整数，最大正方形的边长。

Sample

standard input	standard output

`4 4`
`0 1 1 1`
`1 1 1 0`
`0 1 1 0`
`1 1 0 1`
	`2`

Problem J. Yasaka and squares

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

四方定理众所周知：任意一个正整数n，可以分解为不超过四个整数的平方和。比如$25 = 1^2 + 2^2 + 2^2 + 4^2$,或者$25 = 3^2 + 4^2$。现在给定正整数$n$，Yasaka想知道其能分解出的平方和的总数。

Input

第一行为正整数$t(1 \le t \le 100)$，接下来t行，每行一个正整数$n(1 \le n \le 32768)$。

Output

对于每个方案数n，输出方案总数

Sample

standard input	standard output
1 25	3

Problem K. Haruhi and Kotori

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

$$f_i = \begin{cases} x & i = 1 \\ y & i = 2 \\ af_{i-2} + bf_{i-1} & i \geq 3 \end{cases}$$

Haruhi临走前整了个函数f，Kotori整了个n长数列$C = c_1, c_2, \cdots, c_n$称作Kotonacci，对TA的这个数列进行了$q$次操作，每一次操作选择Kotonacci上连续的一段$c_l \sim c_r$，对于所有的$i(l \leq i \leq r)$，Kotori使得c_i增加f_{i-l+1}

q次操作后，请你给出最后的数列，数列中的每一项对$10^9 + 7$取模

Input

第一行为5个整数$n, x, y, a, b(1 \leq n \leq 10^5, 1 \leq x, y, a, b \leq 10^9)$

第二行为$c_1, c_2, \cdots, c_n$

第三行为一个整数$q(1 \leq q \leq 10^5)$

接下来q行，每行仅包含两个整数l与$r(1 \leq l \leq r \leq n)$，用空格隔开

Output

输出一行n个数，表示经过q次操作后的数列

Sample

`standard input`	`standard output`

5 2 3 4 5
0 0 0 0 0
6
1 1
2 4
3 5
1 5
1 5
1 5
	8 11 74 407 2204

Problem L. Kaguya with points

Input file:	`standard input`	Time limit:	1 second
Output file:	`standard output`	Memory limit:	256 megabytes

给出D维空间的N个点，求曼哈顿距离最大的两个点的曼哈顿距离。两个D维的点$(X_1, X_2, ..., X_D), (Y_1, Y_2, ..., Y_D)$的曼哈顿距离定义为$|X1 - Y1| + |X2 - Y2| + ... + |Xn - Yn|$。

Input

第一行两个整数N, D

接下来N行，每行D个整数描述一个点的坐标。

$2 \le N \le 1000000, 1 \le D \le 4$。

Sample

standard input	standard output

4 2
2 1
1 4
4 5
5 3
	6