Problem A. As Easy As Possible

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

As we know, the NTU Final PK contest usually tends to be pretty hard. Many teams got frustrated when participating NTU Final PK contest. So I decide to make the first problem as "easy" as possible. But how to know how easy is a problem? To make our life easier, we just consider how easy is a string.

Here, we introduce a sane definition of "easiness". The easiness of a string is the maximum times of "easy" as a subsequence of it. For example, the easiness of "eeaseyaesasyy" is 2. Since "easyeasy" is a subsequence of it, but "easyeasyeasy" is too easy.

How to calculate easiness seems to be very easy. So here is a string s consists of only 'e', 'a', 's', and 'y'. Please answer m queries. The i-th query is a interval $[l_i, r_i]$, and please calculate the easiness of $s[l_i..r_i]$.

Input

The first line contains a string s. The second line contains an integer m. Each of following m lines contains two integers l_i, r_i.

- $1 \le |s| \le 10^5$
- $1 \le m \le 10^5$
- $1 \le l_i \le r_i \le |s|$
- s consists of only 'e', 'a', 's', and 'y'

Output

For each query, please output the easiness of that substring in one line.

Examples

standard input	standard output
easy 3 1 4 2 4 1 3	1 0 0
eeaseyaesasyy 4 1 13 2 12 2 10 3 11	2 2 1 0

Problem B. Be Friends

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

There are n lovely lolis in teacher Jiang's class. Each loli has her own personality. For simplicity, we assume that the personality of i-th loli can be represent as an integer p_i.

It is easy to understand that if two lolis have very different personality, they are hard to be direct friends. To formalize this, when the i-th and the j-th loli is a pair of direct friends, there will be a run-in cost of $p_i \oplus p_j$. Here we use $\oplus$ to denote bitwise xor operation.

We say two lolis are indirect friends if there is an "loli path" between them. For example, if $(1, 2), (2, 3), (3, 4)$ are pairs of direct friends, then $(1, 4)$ is a pair of indirect friends.

As a good teacher, Mr. Jiang wants to make all of the lovely lolis to be friends. That is, any two lolis must be either a pair of direct friends of indirect friends. What is the minimum total run-in cost to make this happen?

Input

The first line contains an integer n. The second line contains n integers $p_1, p_2, \ldots, p_n$.

- $1 \le n \le 10^5$
- $0 \le p_i \le 10^9$

Output

Please output the minimum total run-in cost in one line.

Examples

standard input	standard output
3 5 1 4	5
6 1 2 3 5 8 13	20

Problem C. Coprime Heaven

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

The math class is too easy for genius like Nozomi. So she wants to play some single player card game during the class. But since she is a genius, normal single player card game like Solitaire is too easy for her. Thus she creates a hard game called "Coprime Heaven".

There are n cards in this game, and the i-th card is written with an integer i. At the beginning, she picks m lucky positive integers $l_1, l_2, \ldots, l_m$ with $\sum_{i=1}^{m} l_i = n$. Then she tries to partition the cards into m circles, such that each pair of adjacent cards are coprime.

For example, if $n = 5, m = 2, l_1 = 2, l_2 = 3$, we can partition the cards into two circles $\langle 5, 2\rangle$ and $\langle 3, 1, 4\rangle$, since all adjacent pairs $(5, 2), (2, 5), (3, 1), (1, 4), (4, 3)$ are coprime. Note that a circle with only one number will always be valid.

We also want to be as smart as Nozomi, so here are T coprime heaven puzzles for you.

Input

The first line contains an integer T. Each of following T lines contains lucky numbers of one coprime heaven puzzle.

- $1 \leq T \leq 2000$
- $1 \leq m \leq 4$
- $1 \leq l_i \leq 500$
- l_i is sorted in non-decreasing order

Output

For each puzzle, please print a line with "`QQ`" if there is no valid partition. Otherwise, please print a line with "`^_<`". Followed by m lines denote a valid partition. Note that the order of circles should be same as lucky numbers.

Examples

standard input	standard output
4	^_<
1	1
2 3	^_<
4 5 6	2 3
7 8 9 10	5 1 4
	^_<
	1 14 15 8
	6 11 12 5 13
	10 7 4 9 2 3
	QQ

Problem D. Drawing Hell

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

You feel tired after solving many coprime heaven puzzles. Nozomi thinks coprime heaven is also too easy for her now, and single player card game is too boring. So she want to play a interesting two player game called "Drawing Hell" with you.

At the beginning, you and Nozomi draw n lucky points on a blank page of the textbook. You and Nozomi move in turns. Nozomi goes first. In each turn, one can draw a straight segment to connect two lucky points, if it does not cross over any existing segment and/or lucky points. The player who cannot move loses. Note that you can connect a point more than once, and the first game in sample input will finish in three turns.

You and Nozomi will play this game T times during the class. As we know, Nozomi is very smart and will play optimally. Is there any chance you will win the game?

Input

The first line contains an integer T, followed by the input of T games. Each game starts with a line contains an integer n. Each of the following n lines contains two integer x_i, y_i denoting the coordinates of each lucky points.

- $1 \leq T \leq 1000$
- $1 \leq n \leq 1000$
- $0 \leq |x_i|, |y_i| \leq 1000$
- All the lucky points are distinct.

Output

For each game, please output "`OwO`" if you may win even when Nozomi plays optimally. Otherwise, please output "`T^T`".

Examples

standard input	standard output
3 3 0 0 0 4 4 0 4 0 0 0 4 4 0 1 1 4 0 0 0 4 4 0 2 2	T^T OwO T^T

Problem E. Easiest Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

After school, you complete all homeworks in 10 minutes. Though most classes are easy and boring, you are still looking forward to the classes tomorrow. Because you like to playing games with Nozomi. Maybe you have fallen in love with her?

To make the classes tomorrow more interesting, you are researching a cool game called "Knight Garden", and want to share this game with her tomorrow.

The game is played on a $n \times m$ board. There is a (r, s)-knight starts from the upper left corner. In each move, this knight can jump from (x, y) to $(x \pm r, y \pm s)$ or $(x \pm s, y \pm r)$, if the position lies in the board. For example, if a $(1, 2)$-knight is at position $(3, 3)$ of a 5×5 board, the knight has eight possible moves.

We say a (r, s)-knight on a $n \times m$ board is lucky if it can visit all positions on the board. Note that each position can be visited multiple times. You think that just to check whether a knight is lucky is too easy for Nozomi. So you want to know when n, m are given, how many integer pairs (r, s) such that $1 \le r \le s \le \max(n, m)$ and (r, s)-knight is lucky on a $n \times m$ board? As usual, there are T similar questions.

Input

The first line contains an integer T.

Each of the following T lines contains two integers n, m.

- $1 \le T \le 5000$
- $1 \le n, m \le 10^7$

Output

For each question, please output the number of lucky knights on that board.

Examples

standard input	standard output
4 4 4 4 8 8 8 100 100	1 1 4 518

Problem F. Fibonacci of Fibonacci

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

The math class is coming again. You can't wait to tell Nozomi the game "Knight Garden", and she return a puzzle called "Fibonacci of Fibonacci" back to you.

We are all familiar with Fibonacci sequence, which can defined by the recurrence relation as follows.

$$F_0 = 0$$
$$F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2}$$

Since calculating F_n mod 20160519 is too boring for you, she is asking you to calculate F_{F_n} mod 20160519.

Input

The first line contains an integer T.

Each of the following T lines contains an integer n in a single line.

- $1 \le T \le 10000$
- $1 \le n \le 10^9$

Output

For each test case, output an integer F_{F_n} mod 20160519 in a single line.

Examples

standard input	standard output
2 5 6	5 21

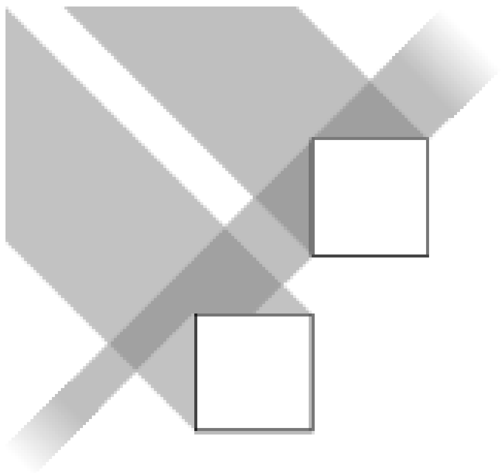
Problem G. Global Warming

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

Global warming becomes an important issue on Meow Planet. As a great scientist, you are trying to predict the temperature in the future. Thus you need to figure out the effect of heating from sunlight.

Since the sun of Meow Planet is far enough, you can assume that the sunlight is parallel and uniformly spread in the space. Meow Planet is a convex polygon. It's obvious that those surface which facing the sun is heating by the sunlight. An 1 Meow meter surface gets 1 Meow joule if the surface is perpendicular to the sunlight. But if the surface is not perpendicular to the sunlight, the energy absorbing may reduce. For example, a surface with $30°$ angle to sunlight gets only 0.5 Meow joule, since the equivalent length which facing to sun remains half. The equivalent length is defined as $-L\cos\theta$, where L is the length of the surface and θ is the angle between the normal vector of the surface and the direction vector of sunlight.

Besides, Meow Planet also has a Meow Moon, which partially reflect sunlight and is a convex polygon too. The Meow Moon reflects sunlight with a specific ratio α, which is a real number between 0 to 1. These reflected light should also be considered, but the energy is multiplied by α. Notice that if an area on Meow Planet is simultaneously lighted by sun and moon, the energy from both should be summed together.



Input

The first line of input contains a integer T ($T \leq 20$), indicating the number of test cases.

Each test case starts with n, m, α, v_x, v_y. n, m are the number of the points of Meow Planet and Meow Moon and α is the reflection ratio of Meow Moon. v_x, v_y is the X and Y component of the direction vector of sunlight. n, m are positive integer and $3 \leq n, m \leq 50000$. α, v_x, v_y are real number, which $0 \leq \alpha \leq 1$ and $-10 \leq v_x, v_y \leq +10$.

Then followed by $n + m$ lines. Each of the first n lines is a point x_i, y_i of the polygon of Meow Planet and Each of the rest m lines is a point x_i, y_i of the polygon of Meow Moon. Both polygon is convex and given in counter-clockwise order. Each point x_i, y_i of both polygon is a real number that $-100000 \leq x_i, y_i \leq +100000$.

All real number are given with at most six digits after the decimal point.

Output

For each test case, only output a line contains the total energy absorbed by the Meow Planet. The answer will be considered correct if the absolute or relative error does not exceed 10^{-6}.

Example

standard input	standard output
1 4 4 0.5 1 -1 0 0 1 0 1 1 0 1 1 1.5 2 1.5 2 2.5 1 2.5	1.590990258

Problem H. Hash Collision

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

As a lazy guy, Shik uses hash heavily when solving problems related to string. Here is a simple hash function implemented in C.

```
int hash(int n, int m, int p, const char *s) {
    int h = 0;
    for (int i = 0; i < n; i++) h = (h * p + s[i]) % m;
    return h;
}
```

If an (unordered) pair of different strings has the same hash values, we say this pair is a "collision pair". Shik claims that he is very lucky and the probability of hash collision is negligible. To verify his claim, you want to calculate the number of collision pairs given n, m, p. Here we only consider the strings consisting of only uppercase letters 'A' to 'Z'. Note that because n is given, the length of string must be exactly n. Since the number could be very large, you only need to output it modulo $10^6 + 3$.

Input

The input contains exactly one line with three integers n, m, p

- $1 \le n \le 10^6$
- $2 \le p < m \le 30000$
- m and p are primes.

Output

Please output the number of collision pairs modulo $10^6 + 3$.

Examples

standard input	standard output
1 3 2	100
2 3 2	75825
21 13 5	142108
50216 9973 131	405787

Problem I. Increasing or Decreasing

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

We all like monotonic things, and solved many problems about that like Longest Increasing Subsequence (LIS). Here is another one which is easier than LIS (in my opinion).

We say an integer is a momo number if its decimal representation is monotonic. For example, 123, 321, 777 and 5566 are momo numbers; But 514, 50216 and 120908 are not.

Please answer m queries. The i-th query is a interval $[l_i, r_i]$, and please calculate the number of momo numbers in it.

Input

The first line contains an integer m.

Each of the following m lines contains two integers l_i, r_i.

- $1 \le m \le 10^5$
- $1 \le l_i \le r_i \le 10^{18}$

Output

For each query, please output the number of momo numbers in that range.

Example

standard input	standard output
2 1 100 100 200	100 48

Problem J. Just Convolution

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

Do you remember the problem "Just Composite" in NTU Final 2014? Here is another one of "Just" series. Of course, This one is easier. I would even give you a correct (but too slow) implementation in C++.

```
struct Con {
    int x;
    Con(int _x) : x(_x) {}
};
Con operator *(Con a, Con b) { return a.x + b.x; }
void operator +=(Con &a, Con b) { if (b.x > a.x) a.x = b.x; }

void convolution(int n, Con *a, Con *b, Con *c) {
    for (int i = 0; i < n; i++) c[i] = 0;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            c[(i + j) % n] += a[i] * b[j];
}
```

Input

The first line contains an integer n. The second line contains n integers $a_0, a_1, \ldots, a_{n-1}$. The third line contains n integers $b_0, b_1, \ldots, b_{n-1}$.

- $1 \le n \le 2 \times 10^5$
- $0 \le a_i, b_i < n$
- a is a permutation of $0, 1, \ldots, n-1$
- b is a permutation of $0, 1, \ldots, n-1$
- The permutations a and b are generated randomly to make our life much more easier.

Output

Please output the value of $c_0, c_1, \ldots, c_{n-1}$ in one line.

Examples

standard input
5 3 4 2 0 1 2 3 0 4 1

standard output
6 6 7 7 8

standard input
10 9 2 0 1 6 3 4 5 8 7 3 8 0 9 7 1 4 2 6 5

standard output
15 17 16 18 16 14 14 15 15 16