Problem A. Hanzo vs. Genji

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

Hanzo just shot an arrow at Genji. Please help him to find out whether that arrow will hit Genji's head directly.

Input

The input contains multiple test cases. Tje first line contains an integer T $(1 \leq T \leq 10^4)$, the number of test cases.

Then T test cases follows. In each test case: the first line contains a vector S, the initial position of Hanzos arrow. The second line contains two positive integers l and r, which means that Hanzo's arrow can be recognized as a cylinder with a radius of r and a height of $2l$. We always use the position of the centroid of the cylinder to indicate the position of the cylinder. Initially the centroid is at S. The third line contains a vector V , the initial velocity of Hanzo's arrow. The fourth line contains a vector G $(|G| > 0)$, the acceleration of gravity. The position of the centroid of the cylinder moves according to V and G. When the arrow is flying, the line connecting the centers of the two circles' surfaces of the cylinder is always the tangent line to the trajectory of the centroid at the point of the centroid. (Consider yourself throwing a telegraph pole if you don't understand the movement of the arrow).

And the last line contains a vector D, the position of Genji's head. We consider Genji's head as a single point. The arrow hits his head when D is on the surface or in the cylinder. Initially, D has no intersection with the cylinder. It is guaranteed that if the arrow will not hit Genji's head, the distance between Genji's head and the arrow will always be no less than 10^{-6}. All the vectors are represented by three numbers which are the three-dimensional cartesion coordinates of that vector.

All the input numbers are integers and their absolute values are bounded by 1000, except for the number of test cases T .

Output

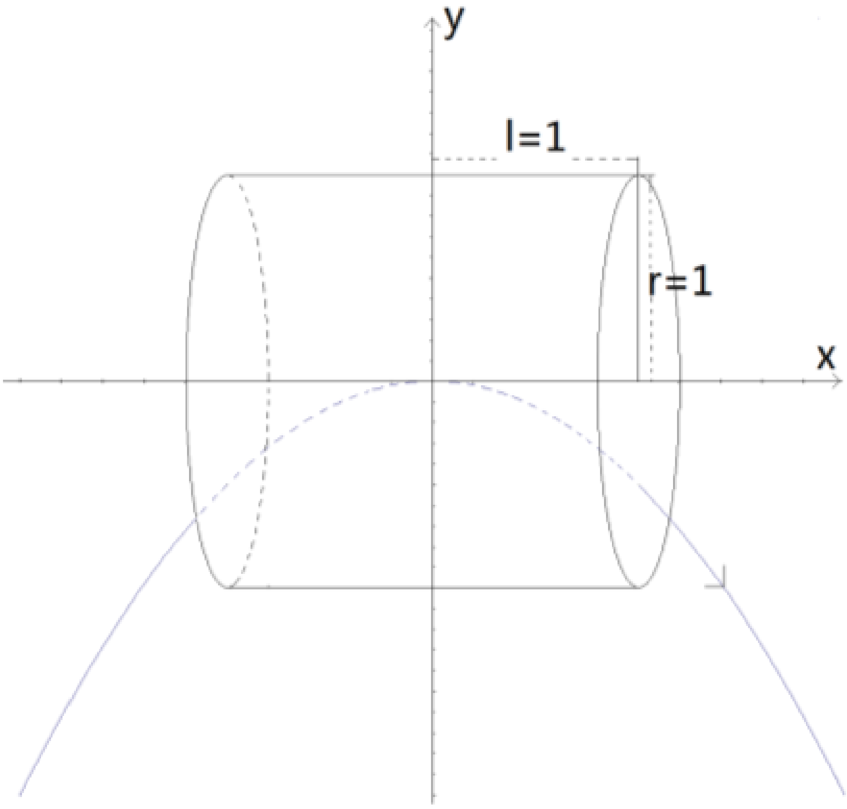
For each test case, output whether the arrow will hit Genji's head. (Please see the sample input and output for details.)

Example

standard input	standard output

2
0 0 0
1 1
1 0 0
0 -1 0
10 0 0
0 0 0
1 1
1 0 0
0 -1 0
2 -2 0
	Case #1: NO
	Case #2: YES

Note



Problem B. Fraction

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

Mr. Frog recently studied how to add two fractions up, and he came up with an evil idea to trouble you by asking you to calculate the result of the formula below:

$$\frac{b_1}{a_1+\frac{b_2}{a_2+\frac{b_3}{\cdots\frac{\cdots}{a_{n-1}+\frac{b_n}{a_n}}}}}$$

As a talent, can you figure out the answer correctly?

Input

The first line contains only one integer T $(1 \leq T \leq 500)$, which indicates the number of test cases.

For each test case, the first line contains only one integer n $(n \leq 8)$. The second line contains n integers: $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10)$. The third line contains n integers: $b_1, b_2, \ldots, b_n$ $(1 \leq b_i \leq 10)$.

Output

For each case, print a line "`Case #x:  p q`", where x is the case number (starting from 1) and p/q indicates the answer as irreducible fraction.

Example

standard input	standard output

1
2
1 1
2 3
	Case #1: 1 2

Problem C. Rotate String

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

Counting strings has been an interesting thing for Mr. Frog since he was 6 years old. Whenever given a string s, Mr. Frog may tell you how many strings in the same length are lexicographically smaller than or equal to s. But now, as an university student, elementary questions may not arouse Mr. Frog's interest, so a harder question appeared.

We call a string t *representative string* if t is the lexicographically smallest string in an equivalence class of strings under rotation (cyclic shift). Given s, Mr. Frog wants to know how many representative strings t, in the same length as s, are lexicographically smaller than or equal to s. In order to be a magician, you may want to solve this challenge quickly to attract Mr. Frog's attention. Only those t consisting of lowercase English letters should be considered.

Input

The first line contains only one integer T ($T \leq 30$), which indicates the number of test cases. For each test case, there is only one line describing the given string s ($1 \leq |s| \leq 100$), s consists of lowercase English letters.

Output

For each test case, output one line “`Case #x:  y`”, where x is the case number (starting from 1) and y is the number of representative strings to this question modulo $10^9 + 7$.

Example

standard input	standard output
2 abb bab	Case #1: 27 Case #2: 651

Note

For the first sample, the representative strings smaller than or equal to “`abb`” are “`aaa`”, “`aab`”, ..., “`aaz`”, “`abb`”. String “`aba`” is not representative, because it's rotation can give string “`aab`”, which is lexicographically less.

Problem D. Triangle

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

Mr. Frog has n sticks, whose lengths are $1, 2, 3, \ldots n$ respectively. Wallice is a bad man, so he does not want Mr. Frog to form a triangle with three of the sticks here. He decides to steal some sticks! Output the minimal number of sticks he should steal so that Mr. Frog cannot form a triangle with any three of the remaining sticks.

Input

The first line contains only one integer T ($T \leq 20$), which indicates the number of test cases. For each test case, there is only one line describing the given integer n ($1 \leq n \leq 10^9$).

Output

For each test case, output one line "`Case #x:  y`", where x is the case number (starting from 1), y is the minimal number of sticks Wallice should steal.

Example

standard input	standard output
3 2 4 6	Case #1: 0 Case #2: 1 Case #3: 2

Problem E. The Fastest Runner Ms. Zhang

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

Ms. Zhang is a journalist from Hong Kong, and she runs very very fast, so everyone calls her "The fastest runner". There are n cities (numbered from 1 to n) and n roads in the country, the roads are undirected. It costs 1 unit of time to go through each road. There exists at least one path between any pair of cities. Now Ms. Zhang can start her interview in city S, visit all of the n cities and finish her interview in city T. Since Ms. Zhang's interview is simple and naive, the interview does not cost any time. Can you help Ms. Zhang to find the optimal S and T that minimise the total traveling time of Ms. Zhang? If there are multiple optimal solutions, output the solution with the smallest S. If there are still multiple solutions, output the solution with the smallest T.

Input

First line of the input contains one integer C — number of the test cases ($1 \leq C \leq 35$).

The first line of each test case contains one integer n ($1 \leq n \leq 10^5$). In the next n lines, each line contains two integers x and y indicating there is a road between city x and city y.

It is guaranteed that there does not exist a road connecting one city to itself or two roads connecting the same pair of cities.

Output

For each test case, output one line "`Case #x: t S T`", where x is the case number(starting from 1), t is the minimal total traveling time, S and T are the indices of starting and ending city.

Example

standard input	standard output

1
10
7 1
5 3
4 5
8 9
6 4
8 6
2 7
10 3
4 1
9 7
	Case #1: 10 1 10

Problem F. Harmonic Value

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

Thee *harmonic value* of the permutation $p_1, p_2, \ldots, p_n$ is

$$\sum_{i=1}^{n-1} gcd(p_i, p_{i+1})$$

Mr. Frog is wondering about the permutation whose harmonic value is the strictly k-th smallest among all the permutations of $[n]$.

Input

The first line contains only one integer T $(1 \leq T \leq 100)$, which indicates the number of test cases. For each test case, there is only one line describing the given integers n and k $(1 \leq 2k \leq n \leq 10^4)$.

Output

For each test case, output one line "`Case #x:`" p_1 p_2 $\ldots$ p_n, where x is the case number (starting from 1) and $p_1, p_2, \ldots p_n$ is the answer. If there is more than one solution, print any of them.

Example

standard input	standard output

`2`
`4 1`
`4 2`
	`Case #1: 4 1 3 2`
	`Case #2: 2 4 1 3`

Problem G. Instability

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

Long long ago, there was a prosperous kingdom which consisted of n cities and every two cites were connected by an undirected road.

However, one day a big monster attacked the kingdom and some roads were destroyed. In order to evaluate the influence brought by the catastrophe, the king wanted to know the instability of his kingdom. Instability is defined as the number of the unstable subset of $\{1, 2, \dots, n\}$. A set S is unstable if and only if there exists a set A such that $A \subseteq S(|A| \geq 3)$ and A is a clique or an independent set, namely that cites in A are pairwise connected directly or they are pairwise disconnected.

Archaeologist has already restored the m roads that were not destroyed by the monster. And they want you to figure out the instability. Since the answer may be tremendously huge, you are only required to write a program that prints the answer modulo $10^9 + 7$.

Input

The first line contains only one integer $T \leq 40$, which indicates the number of test cases. For each test case, the first line contains two integers n $(3 \leq n \leq 50)$ and m $(1 \leq m \leq n(n-1)/2)$, indicating the number of cities and the number of roads.

Then the m lines follow, each containing two integers x and y, indicating there is a road between the city x and the city y.

It is guarenteed that there does not exist a road connecting the same city and there does not exist two same roads.

Output

For each test case, print a line “`Case #x: y`”, where x is the case number (starting from 1) and y is an integer indicating the instability modulo $10^9 + 7$.

Example

standard input	standard output

2
4 3
1 2
2 3
1 3
3 0
	Case #1: 2
	Case #2: 1

Note

- In the first example, $\{1, 2, 3\}$ and $\{1, 2, 3, 4\}$, containing the subset $\{1, 2, 3\}$ which is connected directly, are considered unstable.
- In the second example, $\{1, 2, 3\}$ is considered unstable because they are not pairwise connected directly.

Problem H. Sequence I

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

Mr. Frog has two sequences $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_m$ and a number p. He wants to know the number of positions q such that sequence $b_1, b_2, \ldots, b_m$ is exactly the sequence $a_q, a_{q+p}, a_{q+2p}, \ldots, a_{q+(m-1)p}$, where $q + (m-1)p \leq n$ and $q \geq 1$.

Input

The first line contains only one integer $T \leq 100$, which indicates the number of test cases. Each test case contains three lines.

The first line contains three space-separated integers $1 \leq n \leq 4 \cdot 10^4$, $1 \leq m \leq 4 \cdot 10^4$ and $1 \leq p \leq 4 \cdot 10^4$. The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$). The third line contains m integers $b_1, b_2, \ldots, b_m$ ($1 \leq b_i \leq 10^9$).

It is guaranteed that sum of all n in the input does not exceed $2 \cdot 10^6$.

Output

For each test case, output one line "`Case #x:  y`", where x is the case number (starting from 1) and y is the number of valid q's.

Example

standard input	standard output

`2`
`6 3 1`
`1 2 3 1 2 3`
`1 2 3`
`6 3 2`
`1 3 2 2 3 1`
`1 2 3`
	`Case #1: 2`
	`Case #2: 1`

Problem I. Sequence II

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2.5 seconds
Memory limit:	512 mebibytes

Mr. Frog has an integer sequence of length n, which can be denoted as $a_1, a_2, \ldots, a_n$. There are m queries. In the i-th query, you are given two integers l_i and r_i. Consider the subsequence $a_{l_i}, a_{l_i+1}, a_{l_i+2}, \ldots, a_{r_i}$.

We can denote the positions(the positions according to the original sequence) where an integer appears first in this subsequence as $p_1^{(i)}, p_2^{(i)}, \ldots, p_k^{(i)}$ (in ascending order, i.e., $p_1^{(i)} < p_2^{(i)} < \ldots < p_k^{(i)}$).

Note that k_i is the number of different integers in this subsequence. You should output $p^{(i)}_{\lceil \frac{k_i}{2} \rceil}$ for the i-th query.

Input

Input starts with two integers n ($n \le 2 \cdot 10^5$) and m ($m \le 2 \cdot 10^5$). There are n integers in the next line, which indicate the integers in the sequence (i.e., $a_1, a_2, \ldots, a_n$, $0 \le a_i \le 2 \cdot 10_5$). There are two integers l_i and r_i in the following m lines. However, Mr. Frog thought that this problem was too young too simple so he became angry. He modified each query to ll_i , rr_i ($1 \le ll_i \le n$, $1 \le rr_i \le n$). As a result, the problem became more exciting. We can denote the answers as $ans_1, ans_2, \ldots, ans_m$. Note that for each test case $ans_0 = 0$. You can get the correct input l_i, r_i from what you read (we denote them as ll_i, rr_i) by the following formula:

$$l_i = \min\{(ll_i + ans_{i-1}) \mod n + 1, (rr_i + ans_{i-1}) \mod n + 1\}$$

$$r_i = \max\{(ll_i + ans_{i-1}) \mod n + 1, (rr_i + ans_{i-1}) \mod n + 1\}$$

Output

Print p_1 p_2 $\ldots$ p_m in one line, where p_1, p_2, $\ldots$, p_m is the answer.

Example

standard input	standard output

`5 2`
`3 3 1 5 4`
`2 2`
`4 4`
	`3 3`

`5 2`
`2 5 2 1 2`
`2 3`
`2 4`
	`3 1`

Note

For the first query of the second test case

$(ll_1 + 0) \mod 5 + 1 = 3$

$(rr_1 + 0) \mod 5 + 1 = 4$

$l_1 = \min(3, 4) = 3$

$r_1 = \max(3, 4) = 4$

Thus the subsequence is “2, 1”. The positions where an integer appears first in this subsequence are “3, 4”. Note that $\lceil \frac{k_1}{2} \rceil = 1$. Therefore the answer is 3, which is the first number in “3, 4”.

For the second query of the second test case

$(ll_2 + 3) \mod 5 + 1 = 1$

$(rr_2 + 3) \mod 5 + 1 = 3$

$l_2 = \min(1, 3) = 1$

$r_2 = \max(1, 3) = 3$

Thus the subsequence is “2, 5, 2”. The positions where an integer appears first in this subsequence are “1, 2”. Note that $\lceil \frac{k_2}{2} \rceil = 1$. Therefore the answer is 1, which is the first number in “1, 2”.

Problem J. Ugly Problem

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

Everyone hates ugly problems. You are given a positive integer. You must represent that number by sum of palindromic numbers. A palindromic number is a positive integer such that if you write out that integer as a string in decimal without leading zeros, the string is an palindrome. For example, 1 is a palindromic number and 10 is not.

Input

In the first line of input, there is an integer T $(1 \le T \le 50)$ denoting the number of test cases. For each test case, there is only one line describing the given integer s $(1 \le s \le 10^{1000})$.

Output

For each test case, output "`Case #x:`" on the first line where x is the number of that test case starting from 1. Then output the number of palindromic numbers you used, n, on one line. n must be no more than 50. Then output n lines, each containing one of your palindromic numbers. Their sum must be exactly s.

Example

standard input	standard output
2 18 1000000000000	Case #1: 2 9 9 Case #2: 2 999999999999 1

Problem K. Binary Indexed Tree

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

Recently, Mr. Frog has learned binary indexed tree. Here is the code of adding t to the interval $[1, x]$:

```
void add (int x, int t) {
for (int i = x; i ! = 0; i -= i & (- i))
    a [ i ] += t ;
}
```

If Mr. Frog is required to add t to the interval $[l, r]$, he will $add(r, t)$, and then $add(l - 1, -t)$. The cost of an interval $[l, r]$ is defined as the number of the "really changed point". The "really changed point" is the point whose value is modified by the given code.

For example, in order to add 1 to the interval $[6, 6]$, Mr. Frog will add 1 to the interval $[1, 6]$ ($a[6]$ and $a[4]$ will be added by 1), and add -1 to the interval $[1, 5]$ ($a[5]$ and $a[4]$ will be added by -1).

As the result, $a[6]$ will be added by 1, $a[5]$ will be added by -1, and $a[4]$ will be added by 0. $a[6]$ and $a[5]$ are "really changed point", and the cost is 2.

Mr. Frog wants to calculate the sum of the cost of the interval $[l, r] \subseteq [1, n]$, where l and r are two integers, modulo $10^9 + 7$. Help Mr. Frog solve the problem.

Input

The first line contains only one integer T ($T \leq 10^4$), which indicates the number of test cases. For each test case, it contains an integer n ($1 \leq n \leq 10^{18}$).

Output

For each test case, output one line "`Case #x: y`", where x is the case number (starting from 1), y is the sum of the cost (modulo $10^9 + 7$).

Example

standard input	standard output
3 1 2 3	Case #1: 1 Case #2: 4 Case #3: 10