Problem A. Count the Permutations

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

You are given an array consisting of $n-1$ integers a_i. Count the number of permutations p of n integers from 1 to n, satisfying the following:

- For every $1 \le i \le n-1$, $\max(p_i, p_{i+1}) = a_i$.

Find the answer modulo 998 244 353.

Input

Each test consists of several sets of input data. The first line contains a single integer t ($1 \le t \le 10,000$) — the number of sets of input data. The description of the input data sets follows.

The first line of each data set contains a single integer n ($2 \le n \le 200,000$) — the length of the permutation.

The second line of each data set contains $n-1$ integers $a_1, a_2, \ldots, a_{n-1}$ ($1 \le a_i \le n$) — the array given.

It is guaranteed that the sum of n across all data sets does not exceed $200,000$.

Output

For each set of input data, output a single integer — the number of permutations satisfying the condition modulo 998 244 353.

Example

standard input	standard output
3 3 3 3 4 2 4 3 8 4 8 8 5 7 7 6	2 0 48

Note

In the first set of input data, the two permutations are $[1, 3, 2]$ and $[2, 3, 1]$.

In the second set of input data, it can be shown that there is no permutation of four elements satisfying the condition.

Problem B. Forcefield

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

Captain Ronda is defending a planet where n alien outposts are located. Each outpost is represented as a circle with its center at coordinates (x_i, y_i) and radius r_i. In order to protect them, she wants to deploy a continuous, unbreakable forcefield that will enclose all the outposts.

The forcefield can take any shape, but it must completely surround every outpost without cutting through them. The forcefield must be formed using a single continuous thread of energy that cannot be cut or broken.

Captain Ronda wants to know:

- What is the minimum possible length of the energy thread required to enclose all outposts?
- What is the area enclosed by the resulting forcefield?

You may assume that the outposts do not overlap or touch each other.

Input

The first line contains a single integer n ($2 \le n \le 100$) — the number of outposts.

Each of the next n lines contains three integers x_i, y_i, and r_i ($-1000 \le x_i, y_i \le 1000$, $1 \le r_i \le 100$) — the coordinates of the center and the radius of the i-th outpost.

It is guaranteed that no two outposts intersect or touch each other.

Output

In a single line print two numbers: the minimum length of the energy thread required, and the area enclosed by the resulting forcefield, respectively.

Your answers will be considered correct if the absolute or relative error does not exceed 10^{-6}.

Examples

standard input	standard output
2 0 0 1 99 99 100	670.80969437 33561.73030009
2 10 0 4 -10 0 4	65.13274123 210.26548246
5 0 100 10 -100 -20 10 100 -20 10 -50 40 10 50 40 10	575.24184011 17438.25913572

Problem C. Partial Sum Operations

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

- A *prefix sum array* of an integer array a of length n is an array b of length n such that

$$b_i = a_1 + a_2 + \ldots + a_i \text{ for } 1 \le i \le n.$$

- A *suffix sum array* of an integer array a of length n is an array b of length n such that

$$b_i = a_n + a_{n-1} + \ldots + a_i \text{ for } 1 \le i \le n.$$

- The *normalization* of an integer array a of length n means replacing every element a_i with

$$a_i \leftarrow \max(\min(a_i, 10^{18}), -10^{18})$$

  for $1 \le i \le n$. In other words, every element is *clamped* to the range $[-10^{18}, 10^{18}]$.

You are given an integer array a of length n.

You are allowed to perform any number of the following operations (in any order):

1. **Negate All Elements:** replace every element a_i with $-a_i$.
2. **Prefix Sum on a Subarray:** choose any subarray of a, replace it with its prefix sum array, then normalize the entire array.
3. **Suffix Sum on a Subarray:** choose any subarray of a, replace it with its suffix sum array, then normalize the entire array.

Find the shortest sequence of operations that makes all elements of a non-negative, i.e. $a_i \ge 0$ for every $1 \le i \le n$.

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-1 \le a_i \le 1$) — the elements of the array.

Output

In the first line, print a single integer m — the minimum number of operations required to make all elements of the array a non-negative.

In the next m lines, print the descriptions of the operations:

- For an operation of the first type, output `1`.
- For an operation of the second type, output `2 l r`.
- For an operation of the third type, output `3 l r`.

Here, l and r ($1 \le l \le r \le n$) are the left and right boundaries of the chosen subarray, both boundaries are inclusive in the subarray.

If there are multiple valid sequences of operations, you may print any of them.

Example

standard input	standard output
7 0 0 1 -1 -1 -1 1	2 3 1 3 2 1 7

Note

In the first example, the array a changes twice:

1. after performing the third type of operation with parameters $l = 1$, $r = 3$, the array a becomes equal to $[1, 1, 1, -1, -1, -1, 1]$;
2. after performing the second type of operation with parameters $l = 1$, $r = 7$, the array a becomes equal to $[1, 2, 3, 2, 1, 0, 1]$.

Problem D. Tricolored circle

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

A circular arrangement contains n candles, numbered from 1 to n in order around the circle. Each candle is burning with a flame of one of three colors: red, yellow, or blue. Let the initial colors of the candles be given by a string s of length n, where the i-th character s_i represents the color of the i-th candle.

There is a target configuration given by a string t of length n, where the i-th character t_i represents the desired final color of the i-th candle.

An operation may be performed on any candle whose two neighboring candles (i.e. the $(i-1)$-th and $(i+1)$-th, with indices modulo n ($i-1$ for the first is n and $i+1$ for the n-th is 1)) are burning with different colors. The operation consists of changing the color of the selected candle to any of the three possible colors.

Determine whether it is possible to reach the target configuration from the initial configuration using at most $10 \cdot n$ operations as described. If it is possible, provide a sequence of operations that achieves this. Otherwise, state that it is impossible.

Input

The first line contains one integer n — the number of candles ($3 \le n \le 100\,000$). Two next lines contains strings s and t, constisting of characters «`R`», «`Y`» и «`B`» ($|s|, |t| = n$). Character «`R`» denotes the red color, «`Y`» denotes the yellow color, and «`B`» denotes the blue color.

Output

If no valid sequence of operations exists to reach the target configuration, print -1.

Otherwise, print an integer k — the number of operations performed ($k \le 10 \cdot n$). Then print k lines describing the operations in the order they are to be executed.

Each of the next k lines should contain:

- an integer p_i ($1 \le p_i \le n$) — the index of the candle to be changed,
- and a character $c_i \in \{\texttt{R}, \texttt{Y}, \texttt{B}\}$ — the new color assigned to that candle.

Note: Minimizing the number of operations is not required.

Examples

standard input	standard output
3 RYB YBR	3 2 B 3 R 1 Y
10 RBRBRYRYYY BBYBRYYBYY	6 8 B 7 Y 1 B 2 R 3 Y 2 B
6 YBYBYB BYBYBY	-1
5 YRRBR YRRBR	0

Problem E. Crazy dance

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

You are given a integers $b_0, b_1, \ldots, b_{a-1}$.

If there exists an integer $x \geq 1$, such that for all $d \in \{0, 1, \ldots, a-1\}$ the number of times digit d appears among integers $1, 2, \ldots, x$ written in base a (without leading zeroes) is exactly b_d, you need to print any such integer. Otherwise, report that there are no such integers.

For example, if $a = 3$, and $x = 5$, the numbers in base a are $1, 2, 10, 11, 12$.

Then $b_0 = 1, b_1 = 5, b_2 = 2$.

Input

The first line has number a — the base of the numerical system ($2 \leq a \leq 100\,000$). The second line contains a integer numbers $b_0, b_1, \ldots, b_{a-1}$ ($0 \leq b_i \leq 10^9$).

Output

If there is no suitable x, print -1. Otherwise, print any such x.

Examples

standard input	standard output
10 1 2 1 1 1 1 1 1 1 1	10
2 3 5	4
5 0 0 0 0 0	-1
3 1 3 1	-1

Problem F. Decart the Grasshopper

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

A grasshopper named Decart lives on the Cartesian plane. Today, he urgently needs to get from point (x_0, y_0) to point (x_1, y_1).

Unfortunately, the grasshopper doesn't have time to think, so he strictly decided to jump n times, with the length of the i-th jump being exactly a_i, as he just dreamed of such a sequence of numbers.

Help the grasshopper to devise a route from point (x_0, y_0) to point (x_1, y_1) using jumps of lengths a_i, or confirm that this is impossible.

Input

The first line contains two integers x_0 and y_0 — the coordinates of the starting point $(-1000 \le x_0, y_0 \le 1000)$.

The second line contains two integers x_1 and y_1 — the coordinates of the ending point $(-1000 \le x_1, y_1 \le 1000)$. Note that the starting and ending points of the grasshopper's route may coincide.

The third line contains a single integer n — the number of jumps the grasshopper makes $(1 \le n \le 10^5)$.

The fourth line contains n integers $a_1, a_2 \dots a_n$ — the desired lengths of the jumps $(1 \le a_i \le 1000)$.

Output

If it is impossible to jump from the starting point to the ending point using the given sequence of jumps, output "`Impossible`" on the first line.

Otherwise, output "`Possible`" on the first line, and in the following n lines, output the coordinates of the points. In the i-th line, two real numbers should be output—the coordinates of the point where the grasshopper will land after the i-th jump.

If there are multiple solutions, any of them is allowed. The following quantities must be equal with a relative or absolute error of no more than 10^{-4}:

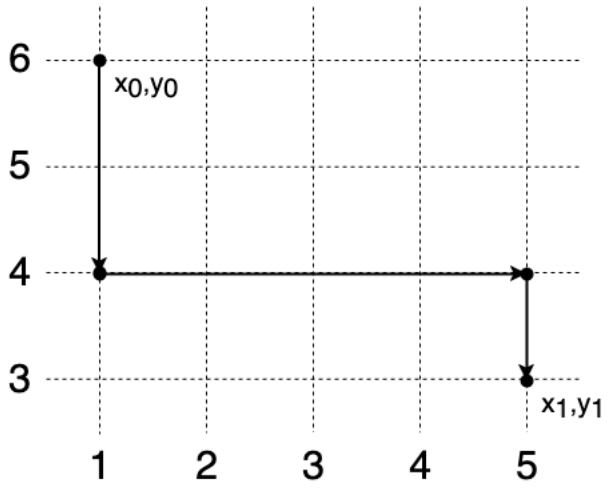
- The distance between the $(i-1)$-th and i-th points in the answer must be equal to a_i;
- The distance between the starting point and the first point in the answer must be equal to a_1;
- The coordinates of the n-th point in the answer must be equal to the coordinates of the ending point.

Examples

standard input	standard output
1 6 5 3 3 2 4 1	Possible 1 4 5 4 5 3
0 0 6 8 5 2 3 4 5 6	Possible -1.20000000 -1.60000000 -3.00000000 -4.00000000 -0.60000000 -0.80000000 2.40000000 3.20000000 6.00000000 8.00000000
-10 10 10 -10 2 5 5	Impossible

Note

One possible route for the grasshopper for the first test from the example is shown in the figure.



Problem G. Heat Exchangers

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

Captain Ronda is conducting experiments again. She has n triangular heat exchangers placed on a field. Each i-th heat exchanger has an input and output coefficient, denoted by kin_i and $kout_i$, respectively. The shape of each exchanger is defined by a triangle with given coordinates of its three vertices.

If two heat exchangers intersect or touch (even at a single point), their inputs are merged. The resulting input coefficient k is calculated as:

$$\frac{1}{k} = \frac{1}{k_1} + \frac{1}{k_2}$$

where k_1 and k_2 are the input coefficients of the exchangers being merged. This merging process continues transitively for any connected group of exchangers.

The field emits heat continuously. Each unit square of the field (with side length 1) emits 1 unit of heat per second. If a point lies beneath one or more exchangers, the heat emitted from that point is multiplied by the final input coefficient of the merged group and by the product of their output coefficients.

Captain Ronda wants to know the total amount of heat emitted per second by all exchangers combined.

Input

The first line contains a single integer n ($1 \le n \le 10$).

Each of the next n lines contains: x_i^1, y_i^1, x_i^2, y_i^2, x_i^3, y_i^3, kin_i, $kout_i$ — the coordinates of the triangle's three vertices followed by its input and output coefficients.

All values are separated by spaces. The coordinates are integers in the range $[-100, 100]$. The coefficients satisfy $0.1 \le kin_i,\ kout_i \le 2.0$. All triangles are non-degenerate.

Output

Output the total amount of heat emitted per second by the entire system. The answer will be accepted if the absolute or relative error does not exceed 10^{-4}.

Examples

standard input	standard output
2 0 0 4 4 8 0 1 1 1 3 7 3 4 -1 1 1	9.51785700
2 -100 100 -100 -100 75 -100 0.5 2 100 100 100 -100 -75 -100 0.5 2	17500.00000000

Problem H. Induced Subgraphs

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

You are given a *tree* with n vertices, numbered from 1 to n. A *tree* is an undirected, connected graph with n vertices and exactly $n-1$ edges and contains no cycles.

Your task is to compute how many *renumberings* (that is, permutations) of the n vertices exist such that for every set of k consecutive numbers $\{i, i+1, \ldots, i+k-1\}$, where $1 \le i \le n-k+1$, the *induced subgraph* formed by these k vertices is *connected*.

A *subgraph induced* by a set of vertices S includes all vertices in S and all edges from the original tree where both endpoints are in S.

Let C be the number of such valid renumberings. You should print C modulo $1\,000\,000\,009$.

Input

The first line contains a single integer n ($2 \le n \le 41$), the number of vertices in the tree.

Each of the next $n-1$ lines contains two integers v_i and u_i ($1 \le v_i, u_i \le n$), representing an undirected edge between vertices v_i and u_i.

The last line contains a single integer k ($1 \le k \le n$), the size of each consecutive group of vertices to be considered.

It is guaranteed that the input describes a tree.

Output

Print a single integer — the number of valid renumberings, modulo $1\,000\,000\,009$.

Examples

standard input	standard output
3 1 2 2 3 2	2
4 1 3 2 3 4 3 3	12
6 6 1 1 2 2 3 3 5 3 4 3	4

Problem I. Magic Spell

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

The great wizard plans to compose a magic spell. The spell consists of runes, which we will denote by integers. The spell that the wizard plans to create is a permutation p consisting of n distinct integers from 1 to n.

To compose the spell, the wizard can use a special magical reference book containing m pages, numbered from 1 to m, each of which contains one rune, an integer from 1 to n. Unfortunately, the wizard is already old and cannot rewrite the runes from the reference book himself; fortunately, he has k students whom he can summon in some order and ask to transcribe fragments of the reference book onto one common long scroll. If the i-th student is summoned, he will sequentially transcribe the runes from the reference book from the l_i-th to the r_i-th page, inclusive. Students can be summoned in any order, but each student can be summoned no more than once.

For example, suppose the reference book has $m = 6$ runes written in the following order: $[2, 1, 3, 1, 2, 4]$, and the wizard has $k = 3$ students with $l_1 = 5$, $r_1 = 6$; $l_2 = 2$, $r_2 = 6$; and $l_3 = 1$, $r_3 = 3$.

If the third student is summoned first and then the first student, they will write the runes onto the scroll in the following order: $[2, 1, 3, 2, 4]$.

However, if the second student is summoned first, then the first, and then the third student, the resulting sequence of runes on the scroll will be $[1, 3, 1, 2, 4, 2, 4, 2, 1, 3]$.

After the runes have been transcribed onto the scroll, the wizard can cut out any continuous segment of the scroll. He wants to summon the minimum number of students so that he can obtain the desired magic spell by cutting out a continuous segment from the resulting scroll.

In the example above, if the spell is the permutation $[1, 3, 2, 4]$, it can be obtained by summoning the third student first and then the first student, and cutting out from the resulting scroll $[2, 1, 3, 2, 4]$ the runes from the 2nd to the 5th.

Help the wizard determine the minimum number of students he must summon and in what order they should be summoned to create the desired spell, if possible.

Input

Each test consists of several sets of input data. The first line contains one integer t ($1 \le t \le 10^5$) — the number of sets of input data. The following describes the sets of input data.

The first line of each set of input data contains three integers n, m, and k ($1 \le n \le 3 \cdot 10^5$, $1 \le m \le 3 \cdot 10^5$, $1 \le k \le 3 \cdot 10^5$) — the length of the permutation p, the number of pages in the book, and the number of students.

The next line contains n distinct integers p_i ($1 \le p_i \le n$) — the permutation p. It is guaranteed that all p_i are pairwise distinct.

The next line contains m integers a_i ($1 \le a_i \le n$) — the runes contained in the book.

The following k lines each contain two integers; the i-th of them contains the numbers l_i and r_i ($1 \le l_i \le r_i \le m$), defining the segment of pages that the i-th student transcribes.

It is guaranteed that the sums of n, m, and k across all sets of input data do not exceed $3 \cdot 10^5$ respectively.

Output

For each set of input data, if it is impossible to summon students in some order such that the permutation p is a subsegment of the sequence of runes written on the scroll, output -1.

Otherwise, output the number c — the minimum number of summoned students.

In the next line, output c numbers — the indices of the students in the order they should be summoned. The students are numbered with integers from 1 to k in the order they are listed in the input. All student indices must be distinct.

If there are multiple optimal answers, output any of them.

Example

standard input	standard output
2 4 6 3 1 3 2 4 2 1 3 1 2 4 5 6 2 6 1 3 3 4 1 3 1 2 2 1 3 1 1 4	2 3 1 -1

Problem J. Least Common Multiple

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

Vasya is a young programmer. He already knows how to solve problems involving graphs, data structures, combinatorics, and dynamic programming. However, he is not yet familiar with number theory. Therefore, he started attending number theory lectures. In one of the first lessons, he learned what the *Least Common Multiple* is.

We define a *multiset* as a collection in which elements may repeat.

The *Least Common Multiple (LCM)* of a multiset S of positive integers is the smallest positive integer that is divisible by every element of the multiset S.

After each lecture, the instructor assigns homework based on the material covered. This time, the homework problem is as follows:

Given two numbers n and k, determine the number of k-element multisets such that their least common multiple is exactly n.

For example, if $n = 6$ and $k = 2$, the valid multisets are: $\{1, 6\}$, $\{2, 3\}$, $\{2, 6\}$, $\{3, 6\}$, $\{6, 6\}$. So, the answer is 5. Note that multisets that differ only in the order of elements are considered the same.

Vasya understood the lecture well and is a very smart boy, so he has already solved the problem. But he is not sure if he did everything correctly. Vasya is asking you to help verify his solution: solve this problem and find the answers for some test cases. Since the answer can be very large, Vasya decided that it is sufficient to find the answer modulo $10^9 + 7$.

Input

The first line of the input file contains two integers n and k ($1 \le n \le 2 \times 10^{18}$, $1 \le k \le 10^6$).

Output

Output one integer — the number of k-element multisets whose LCM is n, modulo $10^9 + 7$.

Examples

standard input	standard output
6 2	5
239 3	3

Problem K. Tennis

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

Alice and Bob decided to play tennis and asked Eve to be the judge. At the beginning, the score was $0 : 0$. Then, a few rounds happened, and after each round, one of the integers in the score was increased by 1. The game ended with the score $a : b$.

Eve was bored so she counted the sum of the gcd's of the players scores after each score change. Gcd is the greatest common divisor of two numbers. For example, the game could go as follows:

- $0 : 0$
- $1 : 0$, $\gcd(1, 0) = 1$
- $2 : 0$, $\gcd(2, 0) = 2$
- $2 : 1$, $\gcd(2, 1) = 1$
- $2 : 2$, $\gcd(2, 2) = 2$
- $2 : 3$, $\gcd(2, 3) = 1$

In that case, Eve would get $1 + 2 + 1 + 2 + 1 = 7$.

After the end of the game, Eve got curious about the minimum number she could potentially get for some sequence of rounds that ends at $a : b$. Your goal is to help her figure out this number.

Input

The first line contains two integers a and b — the final scores of Alice and Bob ($0 \le a, b \le 10^9$).

Output

Output one number — the minimum score which Eve could get for some game with the final scores a and b.

Examples

standard input	standard output
2 1	3
4 6	11
0 0	0
10 10	31

Problem L. Minimum Prefix

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

In the land of Codeonia, the Royal Librarian maintains a collection of ancient scrolls, each with a unique magical incantation written in lowercase Latin letters. The librarian wants to assign labels to each scroll using only the first k letters of their content — called a *k-prefix*. However, to avoid confusion, no two scrolls should share the same k-prefix.

The librarian is allowed to *cyclically shift* the incantations — that is, move some characters from the beginning to the end — before choosing the prefix. The goal is to find the minimum k such that, after applying any cyclic shifts, the k-prefixes of all scrolls are pairwise distinct.

Input

The first line contains one integer n ($2 \le n \le 200$) — the number of scrolls.

Each of the next n lines contains a string consisting of lowercase Latin letters — the incantation on a scroll.

All strings are pairwise distinct.

The total length of all strings does not exceed $2 \cdot 10^5$.

Output

Print a single integer — the minimum k such that, by applying some cyclic shifts to the scrolls, their k-prefixes become pairwise distinct.

Examples

standard input	standard output
4 aaaaa aabac abaaa ababa	2
5 noon oon noo no onn	3

Problem M. Four Villages

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

The kingdom of Byteland consists of n villages, connected by $n-1$ bidirectional roads in such a way that it's possible to reach any village from any other. We will call the distance between two villages $d(v, u)$ the number of roads on a path from village v to village u.

The king of Byteland has four sons. He wants to build a castle for each of them in four different villages. He is afraid that if some two sons live closer to each other than to the other brothers, they will form an alliance against them. So he wants to choose four villages v_1, v_2, v_3, v_4, such that distances $d(v_i, v_j)$ for all pairs $i \neq j$ are the same.

Help the king calculate the number of ways he can choose four villages. The answer may be huge, so calculate it modulo $10^9 + 7$.

Input

The first line contains a single integer t ($1 \leq t \leq 10^5$), the number of test cases. The following t lines contain descriptions of the test cases.

The first line of each test case contains one integer n ($4 \leq n \leq 2 \cdot 10^5$), the number of villages.

Each of the following $n-1$ lines contains two integers u and v, meaning that there is a road between villages u and v.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$

Output

For each test case print a single integer, the number of ways to choose four villages, modulo $10^9 + 7$.

Example

standard input	standard output
2 8 1 2 1 3 1 4 1 5 5 6 5 7 5 8 6 1 2 1 3 1 4 1 5 1 6	2 5