Problem A. Ages of universes

Input file:	`ages.in`
Output file:	`standard output`
Balloon Color:	`Light green`

Rick was teaching Morty about different types of $p-1$ universes, where p is a prime number. The universes are numbered from 1 to $p-1$.

The age of a universe with an index a is the size of the set $\{(a^0 \mod p), (a^1 \mod p), (a^2 \mod p), \cdots (a^k \mod p)\}$, for all non-negative integers k.

Morty was wondering if one of the universes was picked up uniformly at random, what is the expected value of the age of that universe?

Input

The first line of the input contains a single integer T, the number of test cases. Each test case consists of one line, containing a single prime number p, where $1 \le p \le 10^8$.

Output

For each test case, output a single line containing the expected value as a rational number (it is guaranteed that this is always possible), on the format x/y in the reduced form, where x and y are integers.

Example

ages.in	standard output
2 3 5	3/2 11/4

Note

A rational number x/y is in the reduced form if there are no common divisors between x and y other than 1.

Problem B. Binary Meeseeks

Input file:	`binary.in`
Output file:	`standard output`
Balloon Color:	`White`

One day Morty created several Mr. Meeseeks, a blue creature that fulfills one task and then disappears. Morty placed them on a line of N positions represented as a binary string S, where $S_i = 1$ means that there is a Mr. Meeseeks in the i^{th} position and $S_i = 0$ otherwise.

One day Mr. Meeseeks couldn't fulfill their purpose so they started to be problematic. As a result, Rick wanted to take down all Mr. Meeseeks in one shot so he decided to use one of his inventions to perform at most M operations, such that after all operations the distance between the leftmost and rightmost Mr. Meeseeks is minimized. In each operation, a substring of length at most K can be chosen, the substring must start and end with the character '1', then this substring can be moved left or right exactly **one** step, by swapping with the next element in the direction of moving. For example, in '10100010' the substring '101' can be moved right into '01010010'.

Can you help Rick to know the minimum distance?

Input

The first line of the input file contains a single integer T, the number of test cases.
The first line of each test case contains three space-separated integers N, M and K, where $(1 \le K \le N \le 10^5)$ and $(1 \le M \le 10^5)$. The second line contains the string S, it's guaranteed the S contains at least one '1' character.

Output

For each test case, output a single line containing the minimum distance between the leftmost and rightmost Mr. Meeseeks, after performing the operations.

Example

binary.in	standard output
3 8 2 2 11100101 9 4 2 101100111 5 4 2 11111	5 6 4

Note

In the first test case, the string can be transformed using two operations from '11100101' to '11100110' then to '11101100'.

Problem C. Cake for Summer

Input file:	`cake.in`
Output file:	`standard output`
Balloon Color:	`Red`

Summer has brought a cake that's shaped like a grid A of size $N \times M$, where $A_{i,j}$ denotes the number of calories of the cell (i, j). The number of calories in any sub-cake (rectangular subgrid of A) is the minimum number of calories among all its cells.

Since she is making a strict diet, she wants to partition this cake into K sub-cakes, such that they are disjoint and covering the whole original cake.

The partitioning should ensure that the sum of calories of all sub-grids is as minimum as possible. In other words, the partitioning should minimize the summation of the minimum values of calories in each sub-cake. Can you help her to find this minimum summation?

Input

The first line of the input contains a single integer T, the number of test cases.

The first line of each test case contains three space-separated integers N, M, K, followed by N lines, each containing M space-separated integers, the elements of the cake $A_{i,j}$, where $1 \leq A_{i,j} \leq 10^5$ and $1 \leq K \leq N \cdot M \leq 10^5$.

Output

For each test case, output a single line containing a single integer, the minimum summation of all minimum values in the disjoint K subgrids.

Example

cake.in	standard output
2 5 5 5 1 2 3 4 5 2 1 2 3 4 3 2 1 2 3 4 3 2 1 2 5 4 3 2 1 5 5 7 1 1 7 7 7 1 7 7 7 7 7 7 7 7 7 7 7 7 7 1 7 7 7 1 1	5 13

Problem D. Dimension 35-C

Input file:	`dimension35c.in`
Output file:	`standard output`
Balloon Color:	`Yellow`

Rick takes Morty to another dimension, known as Dimension 35-C, which has the perfect conditions for growing 'Mega Trees', which Rick requires for his research. In order to get past an intergalactic custom, they have to answer a puzzle:

Given a list of N distinct integers $A_1, \cdots, A_N$, elements of A are added to a binary search tree, by inserting them one-by-one in a random order (all permutations are equally probable). For each element A_i (for $1 \leq i \leq N$), what is the expected sum of elements in its subtree in the constructed binary search tree?

Rick does research about trees, but he can't solve problems about them, so poor Morty has to solve it, or they can't get past the intergalactic custom. Help Morty to solve this problem!

Input

The first line of the input contains a single integer T, the number of test cases. Each test case consists of a single line containing $N + 1$ space-separated integers, the first one is N and the remaining N numbers form the array $A_1, \cdots, A_N$, where $1 \leq N \leq 10^3$, and $|A_i| \leq 10^9$. The values of the array A are distinct.

Output

For each test case, output N space-separated decimal numbers in a single line, the answers of the expected values corresponding to the values $A_1, \cdots A_N$ in the given order. The answers will be checked with a relative error of 10^{-4}.

Example

dimension35c.in	standard output
3 3 1 2 6 3 6 2 1 3 1 10 100	4.000000 5.500000 7.333333 7.333333 5.500000 4.000000 39.333333 60.500000 105.333333

Note

- A binary search tree is a tree, where each node has a value and at most two children. The left child of a node always has a smaller value than the node, while the right child has a higher value.
- To insert a new value into a binary search tree, the value is compared to the root of the tree. If the new value is larger than the root, then it's inserted into the right subtree, otherwise into the left subtree. This is executed recursively until the insertion reaches a leaf (a node with no children), and then the new value will be inserted as a child of this leaf.
- The binary search tree is not self-balancing, so it doesn't automatically change the tree topology to be a balanced tree.
- In the third test case, there are six possible permutations. For element 10, there are two permutations with subtree sum 111, when 10 is inserted first, and there are two permutations with subtree sum 10 when 10 is inserted last. The remaining two permutations have the subtree sums 110 and 11. Thus the final expected value for the element 10 is $363/6 = 60.50$.

Problem E. Endure is going to space

Input file:	`endure.in`
Output file:	`standard output`
Balloon Color:	`Black`

Rick and Morty have an idea for a super spaceship that can travel from Earth to the Moon in a duration of S seconds, where the distance between the Earth and the Moon here is assumed to be one **light-second**, for simplicity. They are looking forward to traveling to a faraway galaxy at distance Y **light-years**, and they want to get funds for their super spaceship.

They heard about **Endure Capital**, an early-stage investment venture capitalist firm headed by entrepreneurs, which helped previously in funding the development of supersonic airplanes (airplanes that travel long trips more than twice as fast). So they applied to them to get funds to support their trip to that galaxy.

To calculate the funds correctly, they need to know an answer to the question: When can this spaceship reach the faraway galaxy, which is located at distance Y light-years?

Input

The first line of the input contains a single integer T, the number of test cases.

Each test case consists of a single line containing three space-separated items, a string of the name of the galaxy (consisting of at most 30 characters of English alphabet or digits), an integer Y which is the light-years distance of that galaxy and an integer S the number of seconds needed to reach the Moon from Earth, where $1 \le Y, S \le 100$.

Output

For each test case, output a single integer number showing the rounded-down (floored) value of the number of seconds needed to reach this galaxy from the Earth.

Example

endure.in	standard output

`2`
`Andromeda 10 32`
`Cartwheel2 15 56`
	`10091520000`
	`26490240000`

Note

- One year is considered to be exactly 365 day, one day consists of exactly 24 hours and one hour consists of 3,600 seconds.
- 1 light-year is the **distance** traveled by light in one year, and 1 light-second is the **distance** traveled by light in one second.

Problem F. Flip the trees

Input file:	`flip.in`
Output file:	`standard output`
Balloon Color:	`Cyan`

Rick has invented a quantum machine that transfers him between parallel universes, where there are many Ricks and Mortys. Sadly, he got trapped in one of the many parallel universes, and in order to get out, the space police asked Rick to solve this problem:

Given an initially rooted all-white tree of N nodes. Q queries will be given, there are two types of queries:

1. Flip a single node (white $\rightarrow$ black or black $\rightarrow$ white).
2. Ask the minimum number of operations needed to turn the whole subtree rooted at a given node u to be white, wherein one operation you can only flip the colors of the nodes of a whole arbitrary subtree.

Can you help Rick to answer the queries?

Input

The first line of the input contains a single integer T, the number of test cases.

Each test case starts with a line containing two space-separated integers $1 \le N \le 10^5$ and $1 \le Q \le 2 \cdot 10^5$. Followed by $N - 1$ lines where the i^{th} line contains two integers u and v representing an **undirected** edge between the nodes u and v in the tree. The nodes' indices are 1-based, where the root has an index 1.

Followed by Q lines for the queries, where each line contains two integers t and u, where t represents the type of the query (1 represents a query to flip a given node, and 2 represents asking for the minimum number of operations) and u is the node to apply the query on.

Output

For each query of type 2, output a line containing a single integer answering the query.

Example

flip.in	standard output
1 7 6 1 2 3 1 1 4 2 5 2 6 3 7 1 3 1 6 2 1 1 7 2 3 2 1	3 1 2

Note

Please notice the difference between the query of type 1, where you flip a single node and the operations needed for type 2, where you flip whole arbitrary subtrees.

Problem G. Green cube

Input file:	`green.in`
Output file:	`standard output`
Balloon Color:	`Orange`

Rick is really smart, he can turn himself into different objects. He once turned himself into a green pickle. Now, he turned himself into a green cuboid of size $N \times M \times R$, where a diagonal in this cuboid of length k is defined as a sequence (of length k) of triple indices $(a, b, c), (a+1, b+1, c+1), (a+2, b+2, c+2), \cdots, (a+k-1, b+k-1, c+k-1)$ where $1 \le a, b, c$ and $a+k-1 \le N, b+k-1 \le M, c+k-1 \le R$.

Rick then defines a *maximal-diagonal* as a diagonal that can't be extended to a longer diagonal by adding more cells at the beginning or at the end of the diagonal's sequence.

In order for Rick to be able to revert back to a human, Rick asks Morty a puzzle: For all possible positive lengths k of *maximal-diagonals*, output the value of the formula:

$$\bigoplus_{k>0} F(N, M, R, k)$$

where $F(N, M, R, k)$ is the count of all possible *maximal-diagonals* of length k in the given cuboid and $\oplus$ is the bitwise XOR operation.

Can you help Morty to solve this question, to help Rick become human again?

Input

The first line of the input contains a single integer T, the number of test cases. Each test case consists of one line containing three space-separated integers N, M, R where $1 \le N, M, R \le 10^5$.

Output

For each test case, output a line containing a single integer, the value of the formula.

Example

green.in	standard output
3 1 1 1 2 2 2 4 6 9	1 7 60

Note

In the second test case, there is 1 maximal-diagonal of length two, and 6 maximal-diagonals of length one, hence the formula above evaluates to $1 \oplus 6 = 7$

Problem H. Help for Morty the car

Input file:	help.in
Output file:	standard output
Balloon Color:	Silver

Rick has made one of his usual scientific tricks and transformed Morty into a car. Now, Morty (as a car) can be seen as a 1D segment in the 2D plane. The position of Morty is denoted by the position of the midpoint of that segment.

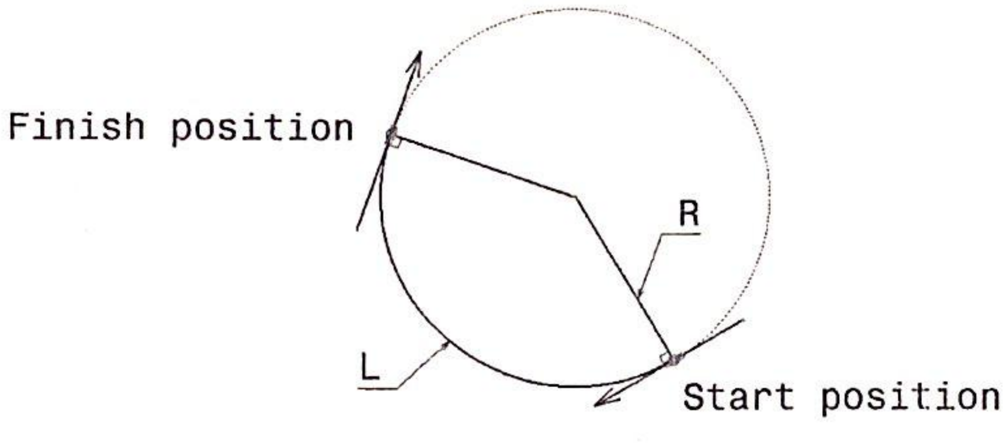
Morty can move in only one possible way, by rotating around some chosen point on the plane (which is not on the straight line of the segment) such that, the segment is always tangential to the circular path Morty is following, as shown in the figure below. The distance he traveled is equal to the length of the arc he traversed (denoted by L below). The point can be chosen freely if it satisfies the specified constraints.

Initially, Morty is located somewhere along the straight line $x = u$, where u is a given integer. Morty's segment and the line are coinciding. In order to cancel the transformation, Morty needs to coincide with the line $x = 0$ and the midpoint should coincide with the origin $(0, 0)$.

You are asked to find a point $p = (u, y)$, where $y \geq 0$, such that if Morty was initially at p, then:

1. With only two rotations around two chosen points (as explained), he can cancel the transformation.
2. The total traveled distance by Morty (after the two rotations) is minimized.

To help Morty, you're asked to output the value y and the corresponding minimum traveled distance.



Input

The first line contains the number of test cases T. Each consists of a line with an integer u ($100 \leq u \leq 10^5$).

Output

For each test case, output a line containing two space-separated decimal numbers: the coordinate y of the starting point and the corresponding minimum distance traveled by Morty. Both numbers should be rounded to their nearest integers, each number will be individually tolerated (because of rounding) with a small absolute difference. You are guaranteed that the minimum length of the path doesn't exceed $2 \cdot 10^5$.

Example

help.in	standard output
1 247	106.0000 341.0000

Problem I. Interdimensional sequences

Input file:	`interdimensional.in`
Output file:	`standard output`
Balloon Color:	`Pink`

Rick was teaching Morty about sequences, how they relate to different dimensions, and how they interact between different dimensions (interdimensional sequence interactions).

He shows three sequences A, B, X, where A is strictly increasing and all are of length N. An interdimensional transformation operation can be done to A by transforming it into a sequence C by lowering the values of A, such that:

1. C should be **non-decreasing**.
2. For all i, it should be $B_i \leq C_i \leq A_i$.

Rick wants to check if Morty understood the interdimensional transformation, so he asked him to find a transformation from A to C with the above constraints, such that it maximizes the formula:

$$\sum_{i=1}^{N}(A_i - C_i) \cdot X_i$$

where X_i can be positive, negative, or zero. Can you help Morty to solve this problem?

Input

The first line of the input file contains a single integer T, the number of test cases. The first line of the test case contains a single integer N (where $1 \leq N \leq 10^5$), the size of the sequences.

Followed by three lines for the sequences A, B, X respectively, each line consists of N space-separated integers. For all $1 \leq i \leq N$ the two constraints $1 \leq B_i \leq A_i \leq 10^6$ and $-10^6 \leq X_i \leq 10^6$ are satisfied.

Output

For each test case, output a single line containing a single integer, the maximum value of the given formula.

Example

interdimensional.in	standard output
1 4 2 4 6 8 1 2 3 4 10 10 -20 10	50

Problem J. Jaguar on the tower

Input file:	`jaguar.in`
Output file:	`standard output`
Balloon Color:	`Green`

Jaguar is a friend of Rick and Morty, who is really brave, and also he likes to jump from high buildings. As usual, Rick used his portal gun to go to a distant galaxy, but he traveled into a strange universe, where it's modeled as a 3D Euclidean space. In this universe, there are N towers, their peaks are modeled as points (x_i, y_i, h_i) and Morty is located at the horizontal position (x, y). (x_i, y_i) are the horizontal positions of the towers and h_i are their heights.

They know that Jaguar is standing on the top of one of the towers and that he might be in danger because he might jump from it, so they want to know on which tower he is.

A tower can't be seen by Morty if there is another tower of the same height or higher in front of it, with respect to Morty's standing position. What's the number of towers that Morty can see?

It's guaranteed that all horizontal positions are distinct, and Morty's position doesn't collide with any tower's horizontal position.

Input

The first line of the input contains an integer T, the number of test cases.

For each test case, the first line contains three space-separated integers N, x, y $(1 \leq N \leq 10^5)$.

Followed by N lines, each contains three space-separated integers x_i, y_i, h_i, where (x_i, y_i) is the horizontal location of the i^{th} tower and h_i is its height. All coordinates x, y are integers in the range $[-10^6, 10^6]$ and the heights are integers in the range $[1, 10^6]$.

Output

For each test case, output a single line containing a single integer, the number of towers that Morty can see if he's located at the position (x, y).

Example

jaguar.in	standard output
1 6 0 0 0 1 4 0 2 7 0 3 5 0 5 7 1 1 1 -1 0 2	4

Problem K. K-th similar password

Input file:	`kth.in`
Output file:	`standard output`
Balloon Color:	`Purple`

Rick is inventing his next really advanced portal gun, that can travel between parallel universes. He has figured out that there are aliens who can decrypt the password mechanism of his new gun. He wants Morty to decide a password for him and ensure that it's safe, where a password is modeled as a string.

Two passwords are similar if they consist of the same set of distinct characters (the counts and order of the characters don't matter). Given a text S and a password P, since the password P is strong and Morty wants something similar to it, then among all the substrings of S that are similar to P, Morty should find the K^{th} minimum lexicographically one, because this is the most guaranteed to not be hacked by the aliens.

Two substrings are considered different if they have a different start or different end in S. In other words, two different substrings might have exactly the same sequence of characters, yet considered different because of the different starting or ending positions.

Can you help Morty to find the K^{th} similar password?

Input

The first line of the input is an integer T, the number of test cases.

Each test case consists of three lines, containing one integer K and two strings S and P, where $1 \leq |S|, |P| \leq 10^5$ and $1 \leq K \leq |S| \cdot (|S|+1)/2$. The strings consist of lower-case English characters.

Output

For each test case, output a line containing the K^{th} lexicographically minimum substring. If K is greater than the number of substrings that are similar to P, then output -1 instead.

Example

kth.in	standard output
6 1 aa aaaaa 2 aa aaaaa 9 aaabaa a 10 aaabaa a 1 aaabaa ba 4 gcggzcgz gggcgg	a a aaa -1 aaab gc

Problem L. Legendary Dog-Master

Input file:	`legendary.in`
Output file:	`standard output`
Balloon Color:	`Gold`

The evolved dog Snowball held Rick's family hostages so that Rick would build him a device allowing him to have even more intellect. This will help Snowball becoming a Legendary Dog-Master and thus be able to lead the dogs on another planet.

Rick gave him a function F and a very large number N, and bet him that if Morty can get the sum of $F(X)$ for all $1 \leq X \leq N$ faster than Snowball, then he has to release Rick's family.

Luckily for Morty, unlike Snowball, he already knew the definition of $F(X)$ as a function that transforms X such that, the digit in position j (from left to right) X_j is replaced by the number $X_j^2 \oplus X_{j-1}^2 \cdots \oplus X_0^2$, where X_0 is the leftmost digit, and $\oplus$ is the bitwise XOR operation. The replacements of the digits are concatenated in the same original order of the digits.

Can you help Morty to calculate $\sum_{X=1}^{N} F(X)$?

Input

The first line of the input contains a single integer T, the number of test cases. For each test case, the input will be a single line containing a single integer $1 \leq N \leq 10^{75}$.

Output

For each test case, output a single line containing the sum of $F(X)$ for all X from 1 to N modulo $10^9 + 7$.

Example

legendary.in	standard output
7 4 10 15 40411 40412 9806 9807	30 296 580 647438681 649054696 588424378 669596110

Note

- For the number 9807, X_0 is considered to be 9 and $F(9807) = 81171732$, where 9 is replaced by 81, 8 is replaced by 17, 0 is replaced by 17, and 7 is replaced by 32.
- For the number 40412, the five digits are replaced respectively by $16, 16, 0, 1, 5$, hence $F(40412) = 1616015$.