Problem A. Hotel for giraffes

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

In a hotel for giraffes, the administration wants to stock up on pillows to meet the needs of its potential guests. It is known that giraffes, depending on the length of their necks, need a stack of pillows (one or more pillows in a stack) from 1 to n centimeters thick. At the same time, the administration wants to do with as few pillows as possible, and among the sets of pillows that meet these requirements, the administration will choose a set with the minimum total thickness so that it occupies the minimum volume in the closet. Help the administration to put together the right set of pillows, which allows to get a stack of any height from 1 to `n` centimeters inclusive.

Input

The input data contains a single integer `n` ($1 \leq n \leq 10^9$) the maximum length of the giraffe's neck.

Output

In a single row, output the thickness of each pillow in this set in random order. If there are several answers, display any of them.

Example

standard input	standard output
`9`	`1 2 3 3`

Note

In the example from the statement, it is necessary to choose such a set from the minimum number of pillows, so that using these pillows it was possible to make a stack of any entire thickness from 1 to 9 cm. Such a set is a set of pillows with a thickness of 1, 2, 3, 3 cm. A stack of thickness 1 2, 3 cm can be folded from one pillow. The remaining net obtained as follows: $4 = 1 + 3$, $5 = 2 + 3$, $6 = 3 + 3$, $7 = 1 + 3 + 3$, $8 = 2 + 3 + 3$, $9 = 1 + 2 + 3 + 3$. Other answer options are possible with the same number of pillows and their total thickness. It is impossible to fulfill the condition of the task using only three pillows.

Problem B. Ledi's new game

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

Lady is playing a new game - "River Battle". This game presents a strip that is `n` cells long and one cell wide. A ship of `k` cells is located somewhere on the field `k`$(k \le n)$.

Input

The first line of input contains an integer `n` $(1 \le n \le 10^9)$. The second input line contains an integer `k` $(1 \le k \le n)$.

Output

Output one number - the number of shots.

Examples

standard input	standard output
4 2	3
4 4	4

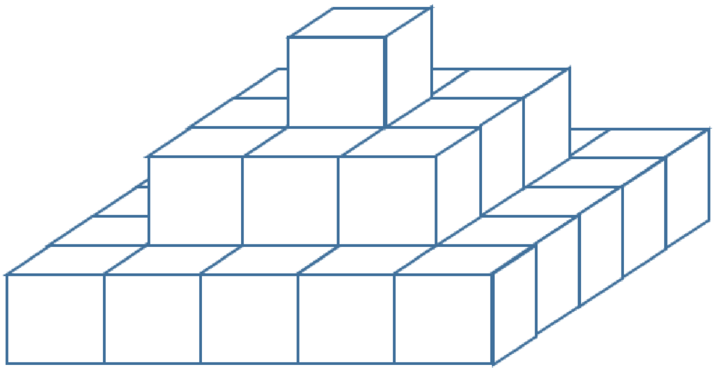
Note

In the first example, the field consists of n = 4 cells, the ship has a length of k = 2. The first shot should be fired at one of the two central cells. If the result is a wound, then the second cell of the ship is in one of the two neighboring cells, and in two shots we are guaranteed to sink the ship. You need 3 shots in total. Two shots are not enough because there is always a chance to miss the first shot.

Problem C. Memorial pyramid

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

Ledi decided to build a pyramid in order to leave a memory of herself for her descendants. The construction plan consists one cubic block at the top of the pyramid (zero layer), 9 blocks below it in the form of a 3x3 platform (first layer), etc., the `N`-th layer of the pyramid consists of `(N×2+1)×(N×2+1)` blocks. To give the structure beauty and durability, the Lady decided to cover the sides of the blocks facing outward (sideways and up) with ceramic tiles (one tile exactly matches the side of the cubic block). That is, for each block, its sides that are visible from the outside are considered. Each such side must be lined with ceramic tiles.



Find out how many tiles are needed to cover a pyramid with layers from 0 to N included. All blocks are the same size.

Input

The first line contains a natural number `N`, which does not exceed 1 000 000 000 - the number of the last layer of the pyramid.

Output

Output one integer - the required number of tiles.

Examples

standard input	standard output
1	25
5	265

Problem D. Representation

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

Ledi loves the number 2 and everything connected with it. For this reason, she also likes the binary number system, except for its only drawback - there is no number with the digit 2. Therefore, she decided to make corrections to the already existing numeral system by creating her own. The Ledi numeral system is exactly the same as binary, except, you guessed it, that you can use the number 2. For example, the number 10 has the following 5 representations in the numeral system:

$1010_{(2)} = 12^3 + 02^2 + 12^1 + 02^0 = 10_{(10)}$

$1002_{(2)} = 12^3 + 02^2 + 02^1 + 22^0 = 10_{(10)}$

$0210_{(2)} = 02^3 + 22^2 + 12^1 + 02^0 = 10_{(10)}$

$0202_{(2)} = 02^3 + 22^2 + 02^1 + 22^0 = 10_{(10)}$

$0122_{(2)} = 02^3 + 12^2 + 22^1 + 22^0 = 10_{(10)}$

Ledi loved her numeral system precisely because one number can be represented in different ways. So she decided to play with the numeral system. Let the number x can be presented as a_x (example, $a_{10} = 5$). Ledi will ask Q questions, each will contain two numbers l_i and r_i ($l_i \leq r_i$), where the i-th of them corresponds to the value $\sum_{j=l_i}^{r_i} a_j = a_{l_i} + a_{l_i+1} + ... + a_{r_i}$. Since the final amount may be too large, Ledi is only interested in the remainder 998244353. Since Ledi is too busy watching a new movie, she asks you to write a program that finds the answers to the Q queries she asks.

Input

The first line of standard input contains an integer `Q` ($1 \leq Q \leq 10^5$), which is equal to the number of requests. Requests are made in the following `Q` lines, the `i`-th line contains numbers l_i and r_i ($1 \leq l_i \leq r_i \leq 10^{18}$), separated by a space.

Output

Output `Q` lines with the answer to the i-th query in the i-th of the line.

Examples

standard input	standard output
3 10 10 1 9 13 21	5 20 40
4 1 1000 1 60000 1 1000000 1 1000000000	29055 19518908 669290218 584912003

Problem E. Universal

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

The IT company has `N` employees. `P` are programmers, `T` – testers. Let's call an employee universal if he is both a programmer and a tester. Find the minimum and maximum possible number of universals in the company.

Input

Three natural numbers are given `N, P, T` ($1 \leq P, T \leq N \leq 10^{18}$) – the number of employees, the number of programmers, and the number of testers, respectively.

Output

Output two non-negative integers - the minimum and maximum possible number of universals.

Examples

standard input	standard output
10 10 10	10 10
10 2 5	0 2

Problem F. Free from squares

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

A natural number `x` called “square-free“, if there is no natural number $y > 1$, for which y^2 divides x without a remainder. The condition of this problem is very simple - given a natural number n, find the number of “square-free“ numbers in the interval $[1, n]$. Since this version is very easy, you need to answer q of these tests. Write a program that, for q tests consisting of a single value of n, finds the number of “square-free“ square numbers less than or equal to n.

Input

The first line of standard input contains a single number `q` ($1 \le q \le 500$) - amount of tests. Each of the following q lines contains the value `n` ($1 \le n \le 10^{14}$) of the corresponding test.

Output

The program should output q lines, each of which contains one integer - the answers for each test.

Example

standard input	standard output
2 4 10	3 7

Note

Numbers without squares, less than or equal to 10, is 1, 2, 3, 5, 6, 7, 10.

Problem G. Graph

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

Given a natural number n and an undirected graph. Let $d(a, b)$ indicate the shortest distance by the number of edges from vertex a to vertex b, if there is a path between a and b, and ∞ if there is no path. We will call the four vertices (u_1, v_1, u_2, v_2) "interesting" if the following conditions are simultaneously fulfilled:

u_1, v_1, u_2, v_2 mutually different

$d(u_1, v_1) = 1$

$d(u_2, v_2) = 1$

$d(u_1, u_2) \neq \infty$

$d(v_1, v_2) \neq \infty$

Write a program that finds the minimum possible value $d(u_1, u_2) + d(v_1, v_2)$ for all "interesting" four vertices.

Input

The first line of input contains a number `n` ($1 \leq n \leq 5000$). Then there are $n - 1$ lines, such that in line i there are $n - 1 - i$ symbols $x_{i(i+1)}, \ldots, x_{in}$, if $x_{ij} = 1$, then there is an edge between vertices i and j (respectively, also between vertices j and i), and if it is equal to 0, then there is no edge between them.

Output

Print the minimum possible sum of distances, or -1 if a quadrilateral of the desired type does not exist.

Example

standard input	standard output
4 111 11 1	2

Problem H. GCD

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Given natural number n.

Find the number of sequences $(a_1, a_2, \ldots, a_k)$ natural numbers, for which:

$1 \le k$

$1 \le a_1, a_2, \ldots, a_k \le n$

For $d_i = GCD(a_1, a_2, \ldots, a_i)$ the inequality $d_1 > d_2 > \cdots > d_k$ is true.

It is proved that there are a finite number of such sequences. Print the answer by modulo $1\,000\,000\,007$.

Input

The first line contains a natural number n.

Output

Print the found number by modulo $1\,000\,000\,007$

Example

standard input	standard output
2	3

Problem I. Common substring

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

Today we will solve the well-known longest common substring (consecutive elements) problem, but with two additional conditions:

1. *Strings are cyclic.*
2. *In some cases, we will look for the longest common substring between more than two strings.*

More formally, we have a set of cyclic strings $s_1, \ldots, s_k$, each of length n. A string p is a substring of a cyclic string t, if there exists a pair of indices $1 \le i, j \le |t|$, such that $t(i, j) = p$. In this problem, we want to find the length of the longest string p, so that it is simultaneously a substring of all cyclic strings $s_1, \ldots, s_k$.

In the above definition $|t|$ indicates the length of string t and defines $t(i, j)$ as a substring, starting at position i and ending at position j, potentially going from the end of string t to its beginning when $i > j$. In other words, $(i, j) = t_i t_{i+1} \ldots t_j$, when $i \le j$, and $t(i, j) = t_i t_{i+1} \ldots t_{|t|} t_1 t_2 t_j$, when $i > j$.

Write a program that, for k cyclic strings, finds the length of the longest common substring. Note that only initialized strings are looped, not the substring we're looking for.

Input

The first line contains two numbers n and k ($1 \le n \le 100\,000, 2 \le k \le 10$) - the length of the strings and their amount. Each of the next k lines also contains strings themselves consisting of lowercase Latin letters.

Output

Output a single integer — the length of the longest common substring.

Example

standard input	standard output
5 2 fabcq bcdda	3

Problem J. Travels

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

Ledi recently learned that a new communications supermarket has opened in Vinnytsia, and as an avid amateur, she cannot miss such an event. So she decided to go straight there, enjoying the gems of the communication system.

A public transport map can be represented as `N` stops, numbered from 1 to `N`, and `M` streets with two-way traffic between them. Each stop can be reached from any other street. More than one street can be placed between one pair of stops. Initially, Ledi is at stop 1 and has to go to stop `N`. There are a total of `K` transport lines numbered from 1 to `K` - some of them trolleybuses and others trams. Each individual street has exactly one line, and each line is a connected part of stops and streets (we can get from each to every stop comprising the line using only that line).

Pass cards in public transport are valid only for a certain line - the price for using the i-th of them is c_i. Once purchased, the card is valid for a whole day and does not need to be purchased again the second time you use the line.

Now Ledi wonders what is the lowest price she will have to pay to get to the opening. In addition, if there are several route options, she will prefer the one with the fewest tram lines.

Help her by writing a program that finds the cheapest route on a given road network.

Input

The first line of standard input contains three natural numbers `N`, `M` and `K` ($2 \leq N \leq 10^5, 1 \leq K \leq M, N \leq M \leq 10^6$) - the number of stops, streets and transport routes. In the next two lines `K` numbers are entered - the first contains the type - 0 for the trolleybus and 1 for the tram, and the second - the prices $c_i (1 \leq c_i \leq 10^9)$ for the lines. The last `M` lines contain 3 positive integers - u, v and t $(1 \leq u, v \leq N, u \neq v, 1 \leq t \leq K)$ - a street with two-way traffic between stops u and v, along which runs the line numbered t.

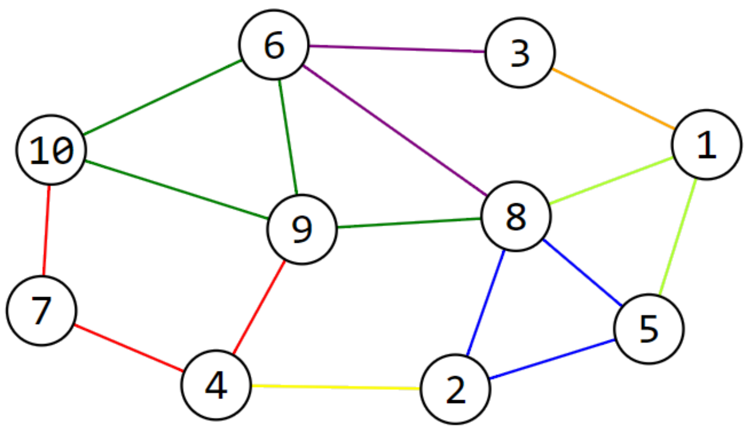
Output

Output the required minimum price and the minimum number of tram lines to be used.

Example

standard input	standard output
10 16 7 1 0 0 1 0 1 0 8 5 3 4 1 2 11 1 5 1 1 8 1 3 1 2 6 3 3 6 8 3 2 8 4 2 5 4 5 8 4 2 4 5 4 9 6 7 4 6 10 7 6 10 9 7 9 8 7 6 9 7 10 6 7	15 2

Note



In the figure, the lines are marked with different colors. It is cheapest to pay for lines 2, 3, 4, 5 and 6. Note that the total price for lines 1, 4, 5 and 6 is also 15, but this route uses 3 tram lines.