A. Find Minimum Operations

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

You are given two integers n and k.

In one operation, you can subtract any power of k from n. Formally, in one operation, you can replace n by $(n - k^x)$ for any non-negative integer x.

Find the minimum number of operations required to make n equal to 0.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The only line of each test case contains two integers n and k ($1 \le n, k \le 10^9$).

Output

For each test case, output the minimum number of operations on a new line.

Standard Input	Standard Output
6 5 2 3 5 16 4 100 3 6492 10 10 1	2 3 1 4 21 10

Note

In the first test case, $n = 5$ and $k = 2$. We can perform the following sequence of operations:

1. Subtract $2^0 = 1$ from 5. The current value of n becomes $5 - 1 = 4$.
2. Subtract $2^2 = 4$ from 4. The current value of n becomes $4 - 4 = 0$.

It can be shown that there is no way to make n equal to 0 in less than 2 operations. Thus, 2 is the answer.

In the second test case, $n = 3$ and $k = 5$. We can perform the following sequence of operations:

1. Subtract $5^0 = 1$ from 3. The current value of n becomes $3 - 1 = 2$.
2. Subtract $5^0 = 1$ from 2. The current value of n becomes $2 - 1 = 1$.
3. Subtract $5^0 = 1$ from 1. The current value of n becomes $1 - 1 = 0$.

It can be shown that there is no way to make n equal to 0 in less than 3 operations. Thus, 3 is the answer.

,

B. Brightness Begins

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

Imagine you have n light bulbs numbered $1, 2, \ldots, n$. **Initially, all bulbs are on**. To *flip* the state of a bulb means to turn it off if it used to be on, and to turn it on otherwise.

Next, you do the following:

- for each $i = 1, 2, \ldots, n$, flip the state of all bulbs j such that j is divisible by $i^\dagger$.

After performing all operations, there will be several bulbs that are still on. Your goal is to make this number exactly k.

Find the smallest suitable n such that after performing the operations there will be exactly k bulbs on. We can show that an answer always exists.

† An integer x is divisible by y if there exists an integer z such that $x = y \cdot z$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The only line of each test case contains a single integer k ($1 \le k \le 10^{18}$).

Output

For each test case, output n — the minimum number of bulbs.

Standard Input	Standard Output
3 1 3 8	2 5 11

Note

In the first test case, the minimum number of bulbs is 2. Let's denote the state of all bulbs with an array, where 1 corresponds to a turned on bulb, and 0 corresponds to a turned off bulb. Initially, the array is $[1, 1]$.

- After performing the operation with $i = 1$, the array becomes $[\underline{0}, \underline{0}]$.
- After performing the operation with $i = 2$, the array becomes $[0, \underline{1}]$.

In the end, there are $k = 1$ bulbs on. We can also show that the answer cannot be less than 2.

In the second test case, the minimum number of bulbs is 5. Initially, the array is $[1, 1, 1, 1, 1]$.

- After performing the operation with $i = 1$, the array becomes $[\underline{0}, \underline{0}, \underline{0}, \underline{0}, \underline{0}]$.
- After performing the operation with $i = 2$, the array becomes $[0, \underline{1}, 0, \underline{1}, 0]$.
- After performing the operation with $i = 3$, the array becomes $[0, 1, \underline{1}, 1, 0]$.
- After performing the operation with $i = 4$, the array becomes $[0, 1, 1, \underline{0}, 0]$.

- After performing the operation with $i = 5$, the array becomes $[0, 1, 1, 0, \underline{1}]$.

In the end, there are $k = 3$ bulbs on. We can also show that the answer cannot be smaller than 5.

C. Bitwise Balancing

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

You are given three non-negative integers b, c, and d.

Please find a non-negative integer $a \in [0, 2^{61}]$ such that $(a \,|\, b) - (a \,\&\, c) = d$, where $|$ and $\&$ denote the bitwise OR operation and the bitwise AND operation, respectively.

If such an a exists, print its value. If there is no solution, print a single integer -1. If there are multiple solutions, print any of them.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^5$). The description of the test cases follows.

The only line of each test case contains three positive integers b, c, and d ($0 \le b, c, d \le 10^{18}$).

Output

For each test case, output the value of a, or -1 if there is no solution. Please note that a must be non-negative and cannot exceed 2^{61}.

Standard Input	Standard Output

`3`
`2 2 2`
`4 2 6`
`10 2 14`
	`0`
	`-1`
	`12`

Note

In the first test case, $(0 \,|\, 2) - (0 \,\&\, 2) = 2 - 0 = 2$. So, $a = 0$ is a correct answer.

In the second test case, no value of a satisfies the equation.

In the third test case, $(12 \,|\, 10) - (12 \,\&\, 2) = 14 - 0 = 14$. So, $a = 12$ is a correct answer.

,

D. Connect the Dots

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

One fine evening, Alice sat down to play the classic game "Connect the Dots", but with a twist.

To play the game, Alice draws a straight line and marks n points on it, indexed from 1 to n. Initially, there are no arcs between the points, so they are all disjoint. After that, Alice performs m operations of the following type:

- She picks three integers a_i, d_i ($1 \le d_i \le 10$), and k_i.
- She selects points $a_i, a_i + d_i, a_i + 2d_i, a_i + 3d_i, \ldots, a_i + k_i \cdot d_i$ and connects each pair of these points with arcs.

After performing all m operations, she wants to know the number of connected components[†] these points form. Please help her find this number.

[†] Two points are said to be in one connected component if there is a path between them via several (possibly zero) arcs and other points.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^5$). The description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \le n \le 2 \cdot 10^5$, $1 \le m \le 2 \cdot 10^5$).

The i-th of the following m lines contains three integers a_i, d_i, and k_i ($1 \le a_i \le a_i + k_i \cdot d_i \le n$, $1 \le d_i \le 10$, $0 \le k_i \le n$).

It is guaranteed that both the sum of n and the sum of m over all test cases do not exceed $2 \cdot 10^5$.

Output

For each test case, output the number of connected components.

Standard Input	Standard Output
3 10 2 1 2 4 2 2 4 100 1 19 2 4 100 3 1 2 5 7 2 6 17 2 31	2 96 61

Note

In the first test case, there are $n = 10$ points. The first operation joins the points 1, 3, 5, 7, and 9. The second operation joins the points 2, 4, 6, 8, and 10. There are thus two connected components: $\{1, 3, 5, 7, 9\}$ and $\{2, 4, 6, 8, 10\}$.

In the second test case, there are $n = 100$ points. The only operation joins the points 19, 21, 23, 25, and 27. Now all of them form a single connected component of size 5. The other 95 points form single-point connected components. Thus, the answer is $1 + 95 = 96$.

In the third test case, there are $n = 100$ points. After the operations, all odd points from 1 to 79 will be in one connected component of size 40. The other 60 points form single-point connected components. Thus, the answer is $1 + 60 = 61$.

,

E. Expected Power

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   256 megabytes
```

You are given an array of n integers $a_1, a_2, \ldots, a_n$. You are also given an array $p_1, p_2, \ldots, p_n$.

Let S denote the random **multiset** (i. e., it may contain equal elements) constructed as follows:

- Initially, S is empty.
- For each i from 1 to n, insert a_i into S with probability $\frac{p_i}{10^4}$. Note that each element is inserted independently.

Denote $f(S)$ as the bitwise XOR of all elements of S. Please calculate the expected value of $(f(S))^2$. Output the answer modulo $10^9 + 7$.

Formally, let $M = 10^9 + 7$. It can be shown that the answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer x that $0 \le x < M$ and $x \cdot q \equiv p \pmod{M}$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^4$). The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 1023$).

The third line of each test case contains n integers $p_1, p_2, \ldots, p_n$ ($1 \le p_i \le 10^4$).

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

Output

For each test case, output the expected value of $(f(S))^2$, modulo $10^9 + 7$.

Standard Input	Standard Output
4 2 1 2 5000 5000 2 1 1 1000 2000 6 343 624 675 451 902 820 6536 5326 7648 2165 9430 5428 1 1 10000	500000007 820000006 280120536 1

Note

In the first test case, $a = [1, 2]$ and each element is inserted into S with probability $\frac{1}{2}$, since $p_1 = p_2 = 5000$ and $\frac{p_i}{10^4} = \frac{1}{2}$. Thus, there are 4 outcomes for S, each happening with the same probability of $\frac{1}{4}$:

- $S = \varnothing$. In this case, $f(S) = 0$, $(f(S))^2 = 0$.
- $S = \{1\}$. In this case, $f(S) = 1$, $(f(S))^2 = 1$.
- $S = \{2\}$. In this case, $f(S) = 2$, $(f(S))^2 = 4$.
- $S = \{1, 2\}$. In this case, $f(S) = 1 \oplus 2 = 3$, $(f(S))^2 = 9$.

Hence, the answer is $0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{4} + 4 \cdot \frac{1}{4} + 9 \cdot \frac{1}{4} = \frac{14}{4} = \frac{7}{2} \equiv 500\,000\,007 \pmod{10^9 + 7}$.

In the second test case, $a = [1, 1]$, a_1 is inserted into S with probability 0.1, while a_2 is inserted into S with probability 0.2. There are 3 outcomes for S:

- $S = \varnothing$. In this case, $f(S) = 0$, $(f(S))^2 = 0$. This happens with probability $(1 - 0.1) \cdot (1 - 0.2) = 0.72$.
- $S = \{1\}$. In this case, $f(S) = 1$, $(f(S))^2 = 1$. This happens with probability $(1 - 0.1) \cdot 0.2 + 0.1 \cdot (1 - 0.2) = 0.26$.
- $S = \{1, 1\}$. In this case, $f(S) = 0$, $(f(S))^2 = 0$. This happens with probability $0.1 \cdot 0.2 = 0.02$.

Hence, the answer is $0 \cdot 0.72 + 1 \cdot 0.26 + 0 \cdot 0.02 = 0.26 = \frac{26}{100} \equiv 820\,000\,006 \pmod{10^9 + 7}$.

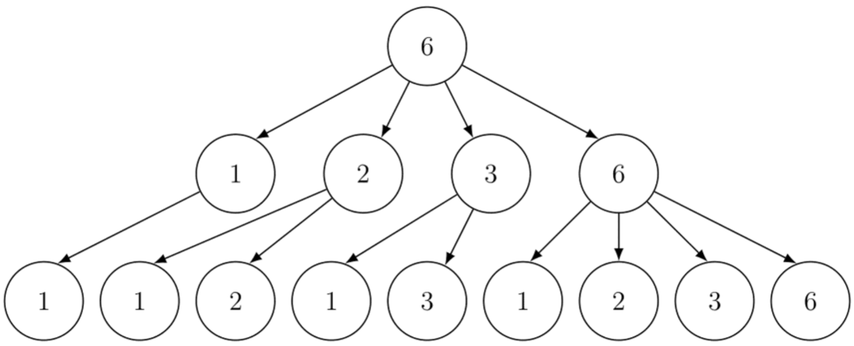
F. Count Leaves

```
Input file:     standard input
Output file:    standard output
Time limit:     4 seconds
Memory limit:   256 megabytes
```

Let n and d be positive integers. We build the *the divisor tree* $T_{n,d}$ as follows:

- The root of the tree is a node marked with number n. This is the 0-th layer of the tree.
- For each i from 0 to $d-1$, for each vertex of the i-th layer, do the following. If the current vertex is marked with x, create its children and mark them with all possible distinct divisors† of x. These children will be in the $(i+1)$-st layer.
- The vertices on the d-th layer are the leaves of the tree.

For example, $T_{6,2}$ (the divisor tree for $n=6$ and $d=2$) looks like this:



Define $f(n,d)$ as the number of leaves in $T_{n,d}$.

Given integers n, k, and d, please compute $\sum_{i=1}^{n} f(i^k, d)$, modulo 10^9+7.

† In this problem, we say that an integer y is a divisor of x if $y \geq 1$ and there exists an integer z such that $x = y \cdot z$.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 10^4$). The description of the test cases follows.

The only line of each test case contains three integers n, k, and d ($1 \leq n \leq 10^9$, $1 \leq k, d \leq 10^5$).

It is guaranteed that the sum of n over all test cases does not exceed 10^9.

Output

For each test case, output $\sum_{i=1}^{n} f(i^k, d)$, modulo 10^9+7.

Standard Input	Standard Output
3 6 1 1	14 1

1 3 3 10 1 2	53

Note

In the first test case, $n = 6$, $k = 1$, and $d = 1$. Thus, we need to find the total number of leaves in the divisor trees $T_{1,1}$, $T_{2,1}$, $T_{3,1}$, $T_{4,1}$, $T_{5,1}$, $T_{6,1}$.

- $T_{1,1}$ has only one leaf, which is marked with 1.
- $T_{2,1}$ has two leaves, marked with 1 and 2.
- $T_{3,1}$ has two leaves, marked with 1 and 3.
- $T_{4,1}$ has three leaves, marked with 1, 2, and 4.
- $T_{5,1}$ has two leaves, marked with 1 and 5.
- $T_{6,1}$ has four leaves, marked with 1, 2, 3, and 6.

The total number of leaves is $1 + 2 + 2 + 3 + 2 + 4 = 14$.

In the second test case, $n = 1$, $k = 3$, $d = 3$. Thus, we need to find the number of leaves in $T_{1,3}$, because $1^3 = 1$. This tree has only one leaf, so the answer is 1.