CODEFORCES β
Sponsored by Telegram

ICM Technex 2018 and Codeforces Round #463 (Div. 1 + Div. 2, combined)

A. Palindromic Supersequence

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a string A. Find a string B, where B is a palindrome and A is a subsequence of B.

A subsequence of a string is a string that can be derived from it by deleting some (not necessarily consecutive) characters without changing the order of the remaining characters. For example, "`cotst`" is a subsequence of "`contest`".

A palindrome is a string that reads the same forward or backward.

The length of string B should be at most 10^4. It is guaranteed that there always exists such string.

You do not need to find the shortest answer, the only restriction is that the length of string B should not exceed 10^4.

Input

First line contains a string A $(1 \le |A| \le 10^3)$ consisting of lowercase Latin letters, where $|A|$ is a length of A.

Output

Output single line containing B consisting of only lowercase Latin letters. You do not need to find the shortest answer, the only restriction is that the length of string B should not exceed 10^4. If there are many possible B, print any of them.

Examples

input Copy
```
aba
```
output
```
aba
```

input Copy
```
ab
```
output
```
aabaa
```

Note

In the first example, "`aba`" is a subsequence of "`aba`" which is a palindrome.

In the second example, "`ab`" is a subsequence of "`aabaa`" which is a palindrome.

B. Recursive Queries

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Let us define two functions f and g on positive integer numbers.

$$f(n) = \text{product of non-zero digits of } n$$
$$g(n) = \begin{cases} n & \text{if } n < 10 \\ g(f(n)) & \text{otherwise} \end{cases}$$

You need to process Q queries. In each query, you will be given three integers l, r and k. You need to print the number of integers x between l and r inclusive, such that $g(x) = k$.

Input

The first line of the input contains an integer Q $(1 \le Q \le 2 \times 10^5)$ representing the number of queries.

Q lines follow, each of which contains 3 integers l, r and k $(1 \le l \le r \le 10^6, 1 \le k \le 9)$.

Output

For each query, print a single line containing the answer for that query.

Examples

input Copy

```
4
22 73 9
45 64 6
47 55 7
2 62 4
```

output

```
1
4
0
8
```

input Copy

```
4
82 94 6
56 67 4
28 59 9
39 74 4
```

output

```
3
1
1
5
```

Note

In the first example:

- $g(33) = 9$ as $g(33) = g(3 \times 3) = g(9) = 9$
- $g(47) = g(48) = g(60) = g(61) = 6$
- There are no such integers between 47 and 55.
- $g(4) = g(14) = g(22) = g(27) = g(39) = g(40) = g(41) = g(58) = 4$

C. Permutation Cycle

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

For a permutation $P[1... N]$ of integers from 1 to N, function f is defined as follows:

$$f(i,j) = \begin{cases} P[i] & \text{if } j = 1 \\ f(P[i], j-1) & \text{otherwise} \end{cases}$$

Let $g(i)$ be the minimum positive integer j such that $f(i, j) = i$. We can show such j always exists.

For given N, A, B, find a permutation P of integers from 1 to N such that for $1 \le i \le N$, $g(i)$ equals either A or B.

Input

The only line contains three integers N, A, B ($1 \le N \le 10^6$, $1 \le A, B \le N$).

Output

If no such permutation exists, output `-1`. Otherwise, output a permutation of integers from 1 to N.

Examples

input

```
9 2 5
```

output

```
6 5 8 3 4 1 9 2 7
```

input

```
3 2 1
```

output

```
1 2 3
```

Note

In the first example, $g(1) = g(6) = g(7) = g(9) = 2$ and $g(2) = g(3) = g(4) = g(5) = g(8) = 5$

In the second example, $g(1) = g(2) = g(3) = 1$

D. Tree

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given a node of the tree with index 1 and with weight 0. Let cnt be the number of nodes in the tree at any instant (initially, cnt is set to 1). Support Q queries of following two types:

- $1\ \mathrm{R}\ \mathrm{W}$: Add a new node (index $cnt+1$) with weight W and add edge between node R and this node.
- $2\ \mathrm{R}\ \mathrm{X}$: Output the maximum length of sequence of nodes which

1. starts with R.
2. Every node in the sequence is an ancestor of its predecessor.
3. Sum of weight of nodes in sequence does not exceed X.
4. For some nodes i, j that are consecutive in the sequence if i is an ancestor of j then $w[i] \geq w[j]$ and there should not exist a node k on simple path from i to j such that $w[k] \geq w[j]$

The tree is rooted at node 1 at any instant.

Note that the queries are given in a modified way.

Input

First line containing the number of queries Q $(1 \leq Q \leq 400000)$.

Let $last$ be the answer for previous query of type 2 (initially $last$ equals 0).

Each of the next Q lines contains a query of following form:

- 1 p q $(1 \leq p, q \leq 10^{18})$: This is query of first type where $R = p \oplus last$ and $W = q \oplus last$. It is guaranteed that $1 \leq R \leq cnt$ and $0 \leq W \leq 10^9$.
- 2 p q $(1 \leq p, q \leq 10^{18})$: This is query of second type where $R = p \oplus last$ and $X = q \oplus last$. It is guaranteed that $1 \leq R \leq cnt$ and $0 \leq X \leq 10^{15}$.

$a \oplus b$ denotes bitwise XOR of a and b.

It is guaranteed that at least one query of type 2 exists.

Output

Output the answer to each query of second type in separate line.

Examples

input Copy

```
6
1 1 1
2 2 0
2 2 1
1 3 0
2 2 0
2 2 2
```

output

```
0
1
1
2
```

input Copy

```
6
1 1 0
2 2 0
2 0 3
1 0 2
2 1 3
2 1 6
```

output

```
2
2
3
2
```

input Copy

```
7
1 1 2
1 2 3
```

```
2 3 3
1 0 0
1 5 1
2 5 0
2 4 0
```

output

```
1
1
2
```

input Copy

```
7
1 1 3
1 2 3
2 3 4
1 2 0
1 5 3
2 5 5
2 7 22
```

output

```
1
2
3
```

Note

In the first example,

$last = 0$

- Query 1: 1 1 1, Node 2 with weight 1 is added to node 1.

- Query 2: 2 2 0, No sequence of nodes starting at 2 has weight less than or equal to 0. $last = 0$

- Query 3: 2 2 1, Answer is 1 as sequence will be $\{2\}$. $last = 1$

- Query 4: 1 2 1, Node 3 with weight 1 is added to node 2.

- Query 5: 2 3 1, Answer is 1 as sequence will be $\{3\}$. Node 2 cannot be added as sum of weights cannot be greater than 1. $last = 1$

- Query 6: 2 3 3, Answer is 2 as sequence will be $\{3, 2\}$. $last = 2$

E. Team Work

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have a team of N people. For a particular task, you can pick any non-empty subset of people. The cost of having x people for the task is x^k.

Output the sum of costs over all non-empty subsets of people.

Input

Only line of input contains two integers N $(1 \le N \le 10^9)$ representing total number of people and k $(1 \le k \le 5000)$.

Output

Output the sum of costs for all non empty subsets modulo $10^9 + 7$.

Examples

input Copy
```
1 1
```
output
```
1
```

input Copy
```
3 2
```
output
```
24
```

Note

In the first example, there is only one non-empty subset $\{1\}$ with cost $1^1 = 1$.

In the second example, there are seven non-empty subsets.

- $\{1\}$ with cost $1^2 = 1$

- $\{2\}$ with cost $1^2 = 1$

- $\{1, 2\}$ with cost $2^2 = 4$

- $\{3\}$ with cost $1^2 = 1$

- $\{1, 3\}$ with cost $2^2 = 4$

- $\{2, 3\}$ with cost $2^2 = 4$

- $\{1, 2, 3\}$ with cost $3^2 = 9$

The total cost is $1 + 1 + 4 + 1 + 4 + 4 + 9 = 24$.

F. Escape Through Leaf

time limit per test: 3 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

You are given a tree with n nodes (numbered from 1 to n) rooted at node 1. Also, each node has two values associated with it. The values for i-th node are a_i and b_i.

You can jump from a node to any node in its subtree. The cost of one jump from node x to node y is the product of a_x and b_y. The total cost of a path formed by one or more jumps is sum of costs of individual jumps. For every node, calculate the minimum total cost to reach any leaf from that node. Pay attention, that root can never be leaf, even if it has degree 1.

Note that you cannot jump from a node to itself.

Input

The first line of input contains an integer n $(2 \le n \le 10^5)$ — the number of nodes in the tree.

The second line contains n space-separated integers $a_1, a_2, \ldots, a_n(-10^5 \le a_i \le 10^5)$.

The third line contains n space-separated integers $b_1, b_2, \ldots, b_n(-10^5 \le b_i \le 10^5)$.

Next $n-1$ lines contains two space-separated integers u_i and v_i $(1 \le u_i, v_i \le n)$ describing edge between nodes u_i and v_i in the tree.

Output

Output n space-separated integers, i-th of which denotes the minimum cost of a path from node i to reach any leaf.

Examples

input Copy

```
3
2 10 -1
7 -7 5
2 3
2 1
```

output

```
10 50 0
```

input Copy

```
4
5 -10 5 7
-8 -80 -3 -10
2 1
2 4
1 3
```

output

```
-300 100 0 0
```

Note

In the first example, node 3 is already a leaf, so the cost is 0. For node 2, jump to node 3 with cost $a_2 \times b_3 = 50$. For node 1, jump directly to node 3 with cost $a_1 \times b_3 = 10$.

In the second example, node 3 and node 4 are leaves, so the cost is 0. For node 2, jump to node 4 with cost $a_2 \times b_4 = 100$. For node 1, jump to node 2 with cost $a_1 \times b_2 = -400$ followed by a jump from 2 to 4 with cost $a_2 \times b_4 = 100$.

G. Palindrome Partition

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given a string s, find the number of ways to split s to substrings such that if there are k substrings $(p_1, p_2, p_3, ..., p_k)$ in partition, then $p_i = p_{k-i+1}$ for all i $(1 \le i \le k)$ and k is even.

Since the number of ways can be large, print it modulo $10^9 + 7$.

Input

The only line of input contains a string s $(2 \le |s| \le 10^6)$ of even length consisting of lowercase Latin letters.

Output

Print one integer, the number of ways of partitioning the string modulo $10^9 + 7$.

Examples

input

```
abcdcdab
```

output

```
1
```

input

```
abbababababbab
```

output

```
3
```

Note

In the first case, the only way to partition the string is $ab|cd|cd|ab$.

In the second case, the string can be partitioned as $ab|b|ab|ab|ab|ab|b|ab$ or $ab|b|abab|abab|b|ab$ or $abbab|ab|ab|abbab$.