

## Codeforces Round #473 (Div. 2)

### A. Mahmoud and Ehab and the even-odd game

time limit per test: 1 second  
 memory limit per test: 256 megabytes  
 input: standard input  
 output: standard output

Mahmoud and Ehab play a game called the even-odd game. Ehab chooses his favorite integer  $n$  and then they take turns, starting from Mahmoud. In each player's turn, he has to choose an integer  $a$  and subtract it from  $n$  such that:

- $1 \leq a \leq n$ .
- If it's Mahmoud's turn,  $a$  has to be even, but if it's Ehab's turn,  $a$  has to be odd.

If the current player can't choose any number satisfying the conditions, he loses. Can you determine the winner if they both play optimally?

#### Input

The only line contains an integer  $n$  ( $1 \leq n \leq 10^9$ ), the number at the beginning of the game.

#### Output

Output "Mahmoud" (without quotes) if Mahmoud wins and "Ehab" (without quotes) otherwise.

#### Examples

<b>input</b>
1
<b>output</b>
Ehab
<b>input</b>
2
<b>output</b>
Mahmoud

#### Note

In the first sample, Mahmoud can't choose any integer  $a$  initially because there is no positive even integer less than or equal to 1 so Ehab wins.

In the second sample, Mahmoud has to choose  $a = 2$  and subtract it from  $n$ . It's Ehab's turn and  $n = 0$ . There is no positive odd integer less than or equal to 0 so Mahmoud wins.

### B. Mahmoud and Ehab and the message

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

Mahmoud wants to send a message to his friend Ehab. Their language consists of  $n$  words numbered from 1 to  $n$ . Some words have the same meaning so there are  $k$  groups of words such that all the words in some group have the same meaning.

Mahmoud knows that the  $i$ -th word can be sent with cost  $a_i$ . For each word in his message, Mahmoud can either replace it with another word of the same meaning or leave it as it is. Can you help Mahmoud determine the minimum cost of sending the message?

The cost of sending the message is the sum of the costs of sending every word in it.

#### Input

The first line of input contains integers  $n$ ,  $k$  and  $m$  ( $1 \leq k \leq n \leq 10^5$ ,  $1 \leq m \leq 10^5$ ) — the number of words in their language, the number of groups of words, and the number of words in Mahmoud's message respectively.

The second line contains  $n$  strings consisting of lowercase English letters of length not exceeding 20 which represent the words. It's guaranteed that the words are **distinct**.

The third line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) where  $a_i$  is the cost of sending the  $i$ -th word.

The next  $k$  lines describe the groups of words of same meaning. The next  $k$  lines each start with an integer  $x$  ( $1 \leq x \leq n$ ) which means that there are  $x$  words in this group, followed by  $x$  integers which represent the indices of words in this group. It's guaranteed that each word appears in exactly one

group.

The next line contains  $m$  space-separated words which represent Mahmoud's message. Each of these words appears in the list of language's words.

## Output

The only line should contain the minimum cost to send the message after replacing some words (maybe none) with some words of the same meaning.

## Examples

input
5 4 4 i loser am the second 100 1 1 5 10 1 1 1 3 2 2 5 1 4 i am the second
output
107

input
5 4 4 i loser am the second 100 20 1 5 10 1 1 1 3 2 2 5 1 4 i am the second
output
116

## Note

In the first sample, Mahmoud should replace the word "second" with the word "loser" because it has less cost so the cost will be  $100+1+5+1=107$ .

In the second sample, Mahmoud shouldn't do any replacement so the cost will be  $100+1+5+10=116$ .

## C. Mahmoud and Ehab and the wrong algorithm

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

Mahmoud was trying to solve the vertex cover problem on trees. The problem statement is:

Given an undirected tree consisting of  $n$  nodes, find the minimum number of vertices that cover all the edges. Formally, we need to find a set of vertices such that for each edge  $(u, v)$  that belongs to the tree, either  $u$  is in the set, or  $v$  is in the set, **or both are in the set**. Mahmoud has found the following algorithm:

- Root the tree at node 1.
- Count the number of nodes at an even depth. Let it be *evenCnt*.
- Count the number of nodes at an odd depth. Let it be *oddCnt*.
- The answer is the minimum between *evenCnt* and *oddCnt*.

The depth of a node in a tree is the number of edges in the shortest path between this node and the root. The depth of the root is 0.

Ehab told Mahmoud that this algorithm is wrong, but he didn't believe because he had tested his algorithm against many trees and it worked, so Ehab asked you to find 2 trees consisting of  $n$  nodes. The algorithm should find an incorrect answer for the first tree and a correct answer for the second one.

## Input

The only line contains an integer  $n$  ( $2 \leq n \leq 10^5$ ), the number of nodes in the desired trees.

## Output

The output should consist of 2 **independent** sections, each containing a tree. The algorithm should find an incorrect answer for the tree in the first section and a correct answer for the tree in the second. If a tree doesn't exist for some section, output "-1" (without quotes) **for that section only**.

If the answer for a section exists, it should contain  $n - 1$  lines, each containing 2 space-separated integers  $u$  and  $v$  ( $1 \leq u, v \leq n$ ), which means that there's an undirected edge between node  $u$  and node  $v$ . If the given graph isn't a tree or it doesn't follow the format, you'll receive wrong answer verdict.

If there are multiple answers, you can print any of them.

Examples

input
2
output
-1 1 2

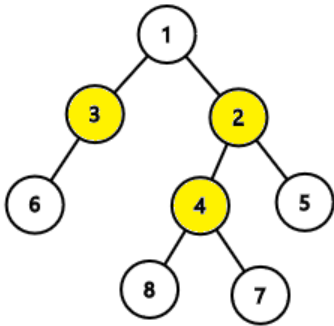
input
8
output
1 2 1 3 2 4 2 5 3 6 4 7 4 8 1 2 1 3 2 4 2 5 2 6 3 7 6 8

Note

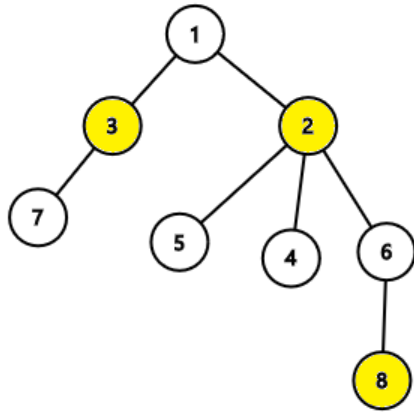
In the first sample, there is only 1 tree with 2 nodes (node 1 connected to node 2). The algorithm will produce a correct answer in it so we printed - 1 in the first section, but notice that we printed this tree in the second section.

In the second sample:

In the first tree, the algorithm will find an answer with 4 nodes, while there exists an answer with 3 nodes like this:



In the second tree, the algorithm will find an answer with 3 nodes which is correct:



## D. Mahmoud and Ehab and another array construction task

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

Mahmoud has an array  $a$  consisting of  $n$  integers. He asked Ehab to find another array  $b$  **of the same length** such that:

- $b$  is lexicographically greater than or equal to  $a$ .
- $b_i \geq 2$ .
- $b$  is pairwise coprime: for every  $1 \leq i < j \leq n$ ,  $b_i$  and  $b_j$  are coprime, i. e.  $GCD(b_i, b_j) = 1$ , where  $GCD(w, z)$  is the greatest common divisor of  $w$  and  $z$ .

Ehab wants to choose a special array so he wants the lexicographically minimal array between all the variants. Can you find it?

An array  $x$  is lexicographically greater than an array  $y$  if there exists an index  $i$  such that  $x_i > y_i$  and  $x_j = y_j$  for all  $1 \leq j < i$ . An array  $x$  is equal to an array  $y$  if  $x_i = y_i$  for all  $1 \leq i \leq n$ .

### Input

The first line contains an integer  $n$  ( $1 \leq n \leq 10^5$ ), the number of elements in  $a$  and  $b$ .

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $2 \leq a_i \leq 10^5$ ), the elements of  $a$ .

### Output

Output  $n$  space-separated integers, the  $i$ -th of them representing  $b_i$ .

### Examples

<b>input</b>
5 2 3 5 4 13
<b>output</b>
2 3 5 7 11
<b>input</b>
3 10 3 7
<b>output</b>
10 3 7

### Note

Note that in the second sample, the array is already pairwise coprime so we printed it.

## E. Mahmoud and Ehab and the xor-MST

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

Ehab is interested in the bitwise-xor operation and the special graphs. Mahmoud gave him a problem that combines both. He has a complete graph consisting of  $n$  vertices numbered from  $0$  to  $n - 1$ . For all  $0 \leq u < v < n$ , vertex  $u$  and vertex  $v$  are connected with an undirected edge that has weight  $u \oplus v$  (where  $\oplus$  is the bitwise-xor operation). Can you find the weight of the minimum spanning tree of that graph?

You can read about complete graphs in [https://en.wikipedia.org/wiki/Complete\\_graph](https://en.wikipedia.org/wiki/Complete_graph)

You can read about the minimum spanning tree in [https://en.wikipedia.org/wiki/Minimum\\_spanning\\_tree](https://en.wikipedia.org/wiki/Minimum_spanning_tree)

The weight of the minimum spanning tree is the sum of the weights on the edges included in it.

**Input**

The only line contains an integer  $n$  ( $2 \leq n \leq 10^{12}$ ), the number of vertices in the graph.

**Output**

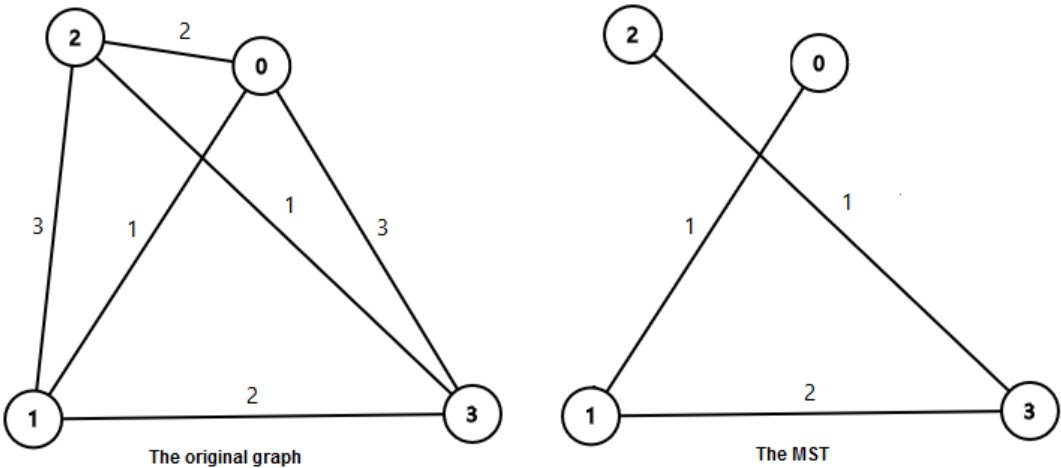
The only line contains an integer  $x$ , the weight of the graph's minimum spanning tree.

**Example**

input
4
output
4

**Note**

In the first sample:



The weight of the minimum spanning tree is  $1+2+1=4$ .

F. Mahmoud and Ehab and yet another xor task

time limit per test: 1 second  
memory limit per test: 512 megabytes  
input: standard input  
output: standard output

Ehab has an array  $a$  of  $n$  integers. He likes the bitwise-xor operation and he likes to bother Mahmoud so he came up with a problem. He gave Mahmoud  $q$  queries. In each of them, he gave Mahmoud 2 integers  $l$  and  $x$ , and asked him to find the number of subsequences of the first  $l$  elements of the array such that their bitwise-xor sum is  $x$ . Can you help Mahmoud answer the queries?

A subsequence can contain elements that are not neighboring.

**Input**

The first line contains integers  $n$  and  $q$  ( $1 \leq n, q \leq 10^5$ ), the number of elements in the array and the number of queries.

The next line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i < 2^{20}$ ), the elements of the array.

The next  $q$  lines, each contains integers  $l$  and  $x$  ( $1 \leq l \leq n, 0 \leq x < 2^{20}$ ), representing the queries.

**Output**

For each query, output its answer modulo  $10^9 + 7$  in a new line.

**Examples**

input
5 5

```
0 1 2 3 4
4 3
2 0
3 7
5 7
5 8
```

**output**

```
4
2
0
4
0
```

**input**

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

**output**

```
4
2
```

### Note

The bitwise-xor sum of the empty set is 0 and the bitwise-xor sum of a set containing one element is that element itself.