Codeforces Round #196 (Div. 1)

A. Quiz

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

Manao is taking part in a quiz. The quiz consists of n consecutive questions. A correct answer gives one point to the player. The game also has a counter of consecutive correct answers. When the player answers a question correctly, the number on this counter increases by 1. If the player answers a question incorrectly, the counter is reset, that is, the number on it reduces to 0. If after an answer the counter reaches the number k, then it is reset, and the player's score is doubled. Note that in this case, first 1 point is added to the player's score, and then the total score is doubled. At the beginning of the game, both the player's score and the counter of consecutive correct answers are set to zero.

Manao remembers that he has answered exactly m questions correctly. But he does not remember the order in which the questions came. He's trying to figure out what his minimum score may be. Help him and compute the remainder of the corresponding number after division by 1000000009 $(10^9 + 9)$.

Input
The single line contains three space-separated integers n, m and k ($2 \le k \le n \le 10^9$; $0 \le m \le n$).

Output
Print a single integer — the remainder from division of Manao's minimum possible score in the quiz by 1000000009 $(10^9 + 9)$.

Examples

input
5 3 2
output
3

input
5 4 2
output
6

Note
Sample 1. Manao answered 3 questions out of 5, and his score would double for each two consecutive correct answers. If Manao had answered the first, third and fifth questions, he would have scored as much as 3 points.

Sample 2. Now Manao answered 4 questions. The minimum possible score is obtained when the only wrong answer is to the question 4.

Also note that you are asked to minimize the score and not the remainder of the score modulo 1000000009. For example, if Manao could obtain either 2000000000 or 2000000020 points, the answer is 2000000000 *mod* 1000000009, even though 2000000020 *mod* 1000000009 is a smaller number.

B. Book of Evil

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

Paladin Manao caught the trail of the ancient Book of Evil in a swampy area. This area contains n settlements numbered from 1 to n. Moving through the swamp is very difficult, so people tramped exactly $n - 1$ paths. Each of these paths connects some pair of settlements and is bidirectional. Moreover, it is possible to reach any settlement from any other one by traversing one or several paths.

The *distance* between two settlements is the minimum number of paths that have to be crossed to get from one settlement to the other one. Manao knows that the Book of Evil has got a damage range d. This means that if the Book of Evil is located in some settlement, its damage (for example, emergence of ghosts and werewolves) affects other settlements at distance d or less from the settlement where the Book resides.

Manao has heard of m settlements affected by the Book of Evil. Their numbers are $p_1, p_2, \ldots, p_m$. Note that the Book may be affecting other settlements as well, but this has not been detected yet. Manao wants to determine which settlements may contain the Book. Help him with this difficult task.

Input

The first line contains three space-separated integers n, m and d ($1 \le m \le n \le 100000$; $0 \le d \le n - 1$). The second line contains m distinct space-separated integers $p_1, p_2, \ldots, p_m$ ($1 \le p_i \le n$). Then $n - 1$ lines follow, each line describes a path made in the area. A path is described by a pair of space-separated integers a_i and b_i representing the ends of this path.

Output

Print a single number — the number of settlements that may contain the Book of Evil. It is possible that Manao received some controversial information and there is no settlement that may contain the Book. In such case, print 0.

Examples

input
6 2 3 1 2 1 5 2 3 3 4 4 5 5 6

output
3

Note

Sample 1. The damage range of the Book of Evil equals 3 and its effects have been noticed in settlements 1 and 2. Thus, it can be in settlements 3, 4 or 5.

C. Divisor Tree

time limit per test: 0.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A *divisor tree* is a rooted tree that meets the following conditions:

- Each vertex of the tree contains a positive integer number.
- The numbers written in the leaves of the tree are prime numbers.
- For any inner vertex, the number within it is equal to the product of the numbers written in its children.

Manao has n distinct integers $a_1, a_2, \ldots, a_n$. He tries to build a divisor tree which contains each of these numbers. That is, for each a_i, there should be at least one vertex in the tree which contains a_i. Manao loves compact style, but his trees are too large. Help Manao determine the minimum possible number of vertices in the divisor tree sought.

Input

The first line contains a single integer n ($1 \le n \le 8$). The second line contains n distinct space-separated integers a_i ($2 \le a_i \le 10^{12}$).

Output

Print a single integer — the minimum number of vertices in the divisor tree that contains each of the numbers a_i.

Examples

input
2 6 10
output
7

input
4 6 72 8 4
output
12

input
1 7
output
1

Note

Sample 1. The smallest divisor tree looks this way:

Sample 2. In this case you can build the following divisor tree:

Sample 3. Note that the tree can consist of a single vertex.

D. GCD Table

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

Consider a table G of size $n \times m$ such that $G(i, j) = GCD(i, j)$ for all $1 \le i \le n, 1 \le j \le m$. $GCD(a, b)$ is the greatest common divisor of numbers a and b.

You have a sequence of positive integer numbers $a_1, a_2, ..., a_k$. We say that this sequence occurs in table G if it coincides with consecutive elements in some row, starting from some position. More formally, such numbers $1 \le i \le n$ and $1 \le j \le m - k + 1$ should exist that $G(i, j + l - 1) = a_l$ for all $1 \le l \le k$.

Determine if the sequence a occurs in table G.

Input

The first line contains three space-separated integers n, m and k ($1 \le n, m \le 10^{12}$; $1 \le k \le 10000$). The second line contains k space-separated integers $a_1, a_2, ..., a_k$ ($1 \le a_i \le 10^{12}$).

Output

Print a single word "YES", if the given sequence occurs in table G, otherwise print "NO".

Examples

input
100 100 5 5 2 1 2 1
output
YES

input
100 8 5 5 2 1 2 1
output
NO

input
100 100 7 1 2 3 4 5 6 7
output
NO

Note

Sample 1. The tenth row of table G starts from sequence {1, 2, 1, 2, 5, 2, 1, 2, 1, 10}. As you can see, elements from fifth to ninth coincide with sequence a.

Sample 2. This time the width of table G equals 8. Sequence a doesn't occur there.

E. Optimize!

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

Manao is solving a problem with the following statement:

He came up with a solution that produces the correct answers but is too slow. You are given the pseudocode of his solution, where the function getAnswer calculates the answer to the problem:

```
getAnswer(a[1..n], b[1..len], h)
  answer = 0
  for i = 1 to n-len+1
    answer = answer + f(a[i..i+len-1], b, h, 1)
  return answer

f(s[1..len], b[1..len], h, index)
  if index = len+1 then
    return 1
  for i = 1 to len
    if s[index] + b[i] >= h
      mem = b[i]
      b[i] = 0
      res = f(s, b, h, index + 1)
      b[i] = mem
      if res > 0
        return 1
  return 0
```

Your task is to help Manao optimize his algorithm.

Input

The first line contains space-separated integers n, len and h ($1 \le len \le n \le 150000$; $1 \le h \le 10^9$). The second line contains len space-separated integers $b_1, b_2, \ldots, b_{len}$ ($1 \le b_i \le 10^9$). The third line contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Output

Print a single number — the answer to Manao's problem.

Examples

input
5 2 10 5 3 1 8 5 5 7
output
2