Educational Codeforces Round 12

A. Buses Between Cities

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

Buses run between the cities A and B, the first one is at 05:00 AM and the last one departs not later than at 11:59 PM. A bus from the city A departs every a minutes and arrives to the city B in a t_a minutes, and a bus from the city B departs every b minutes and arrives to the city A in a t_b minutes.

The driver Simion wants to make his job diverse, so he counts the buses going towards him. Simion doesn't count the buses he meet at the start and finish.

You know the time when Simion departed from the city A to the city B. Calculate the number of buses Simion will meet to be sure in his counting.

Input

The first line contains two integers a, t_a $(1 \le a, t_a \le 120)$ — the frequency of the buses from the city A to the city B and the travel time. Both values are given in minutes.

The second line contains two integers b, t_b $(1 \le b, t_b \le 120)$ — the frequency of the buses from the city B to the city A and the travel time. Both values are given in minutes.

The last line contains the departure time of Simion from the city A in the format `hh:mm`. It is guaranteed that there are a bus from the city A at that time. Note that the hours and the minutes are given with exactly two digits.

Output

Print the only integer z — the number of buses Simion will meet on the way. Note that you should not count the encounters in cities A and B.

Examples

input
10 30 10 35 05:20
output
5

input
60 120 24 100 13:00
output
9

Note

In the first example Simion departs form the city A at 05:20 AM and arrives to the city B at 05:50 AM. He will meet the first 5 buses from the city B that departed in the period [05:00 AM - 05:40 AM]. Also Simion will meet a bus in the city B at 05:50 AM, but he will not count it.

Also note that the first encounter will be between 05:26 AM and 05:27 AM (if we suggest that the buses are go with the sustained speed).

B. Shopping

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

Ayush is a cashier at the shopping center. Recently his department has started a "`click and collect`" service which allows users to shop online.

The store contains k items. n customers have already used the above service. Each user paid for m items. Let a_{ij} denote the j-th item in the i-th person's order.

Due to the space limitations all the items are arranged in one single row. When Ayush receives the i-th order he will find one by one all the items a_{ij} $(1 \le j \le m)$ in the row. Let $pos(x)$ denote the position of the item x in the row at the moment of its collection. Then Ayush takes time equal to $pos(a_{i1}) + pos(a_{i2}) + ... + pos(a_{im})$ for the i-th customer.

When Ayush accesses the x-th element he keeps a new stock in the front of the row and takes away the x-th element. Thus the values are updating.

Your task is to calculate the total time it takes for Ayush to process all the orders.

You can assume that the market has endless stock.

Input

The first line contains three integers n, m and k $(1 \le n, k \le 100, 1 \le m \le k)$ — the number of users, the number of items each user wants to buy and the total number of items at the market.

The next line contains k distinct integers p_l $(1 \le p_l \le k)$ denoting the initial positions of the items in the store. The items are numbered with integers from 1 to k.

Each of the next n lines contains m distinct integers a_{ij} $(1 \le a_{ij} \le k)$ — the order of the i-th person.

Output

Print the only integer t — the total time needed for Ayush to process all the orders.

Example

input
2 2 5 3 4 1 2 5 1 5 3 1
output
14

Note

Customer 1 wants the items 1 and 5.

$pos(1) = 3$, so the new positions are: $[1, 3, 4, 2, 5]$.

$pos(5) = 5$, so the new positions are: $[5, 1, 3, 4, 2]$.

Time taken for the first customer is $3 + 5 = 8$.

Customer 2 wants the items 3 and 1.

$pos(3) = 3$, so the new positions are: $[3, 5, 1, 4, 2]$.

$pos(1) = 3$, so the new positions are: $[1, 3, 5, 4, 2]$.

Time taken for the second customer is $3 + 3 = 6$.

Total time is $8 + 6 = 14$.

Formally $pos(x)$ is the index of x in the current row.

C. Simple Strings

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

`zscoder` loves simple strings! A string t is called simple if every pair of adjacent characters are distinct. For example `ab`, `aba`, `zscoder` are simple whereas `aa`, `add` are not simple.

`zscoder` is given a string s. He wants to change a minimum number of characters so that the string s becomes simple. Help him with this task!

Input

The only line contains the string s ($1 \le |s| \le 2 \cdot 10^5$) — the string given to `zscoder`. The string s consists of only lowercase English letters.

Output

Print the simple string s' — the string s after the minimal number of changes. If there are multiple solutions, you may output any of them.

Note that the string s' should also consist of only lowercase English letters.

Examples

input
aab
output
bab

input
caaab
output
cabab

input
zscoder
output
zscoder

D. Simple Subset

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

A tuple of positive integers $\{x_1, x_2, ..., x_k\}$ is called simple if for all pairs of positive integers (i, j) $(1 \le i < j \le k)$, $x_i + x_j$ is a prime.

You are given an array a with n positive integers $a_1, a_2, ..., a_n$ (not necessary distinct). You want to find a simple subset of the array a with the maximum size.

A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself.

Let's define a subset of the array a as a tuple that can be obtained from a by removing some (possibly all) elements of it.

Input

The first line contains integer n $(1 \le n \le 1000)$ — the number of integers in the array a.

The second line contains n integers a_i $(1 \le a_i \le 10^6)$ — the elements of the array a.

Output

On the first line print integer m — the maximum possible size of simple subset of a.

On the second line print m integers b_l — the elements of the simple subset of the array a with the maximum size.

If there is more than one solution you can print any of them. You can print the elements of the subset in any order.

Examples

input
2 2 3
output
2 3 2

input
2 2 2
output
1 2

input
3 2 1 1
output
3 1 1 2

input
2 83 14
output
2 14 83

E. Beautiful Subarrays

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

One day, ZS the Coder wrote down an array of integers a with elements $a_1, a_2, ..., a_n$.

A subarray of the array a is a sequence $a_l, a_{l+1}, ..., a_r$ for some integers (l, r) such that $1 \le l \le r \le n$. ZS the Coder thinks that a subarray of a is beautiful if the bitwise xor of all the elements in the subarray is at least k.

Help ZS the Coder find the number of beautiful subarrays of a!

Input

The first line contains two integers n and k ($1 \le n \le 10^6, 1 \le k \le 10^9$) — the number of elements in the array a and the value of the parameter k.

The second line contains n integers a_i ($0 \le a_i \le 10^9$) — the elements of the array a.

Output

Print the only integer c — the number of beautiful subarrays of the array a.

Examples

input
3 1 1 2 3
output
5

input
3 2 1 2 3
output
3

input
3 3 1 2 3
output
2

F. Four Divisors

time limit per test: 10 seconds
memory limit per test: 768 megabytes
input: standard input
output: standard output

If an integer a is divisible by another integer b, then b is called the divisor of a.

For example: 12 has positive 6 divisors. They are 1, 2, 3, 4, 6 and 12.

Let's define a function $D(n)$ — number of integers between 1 and n (inclusive) which has exactly four positive divisors.

Between 1 and 10 only the integers 6, 8 and 10 has exactly four positive divisors. So, $D(10) = 3$.

You are given an integer n. You have to calculate $D(n)$.

Input
The only line contains integer n ($1 \le n \le 10^{11}$) — the parameter from the problem statement.

Output
Print the only integer c — the number of integers between 1 and n with exactly four divisors.

Examples

input
10
output
3

input
20
output
5