Birla Institute of Technology and Science, Pilani
Hyderabad Campus

CS F211 Data Structures and Algorithms
Lab 8: Maps, Sets and Strings

Allowed languages: **C, C++**

General Tips

- Indent your code appropriately and use proper variable names. These increase readability and writability of the code. Also, use comments wherever necessary.
- Use a proper IDE or text editors like Sublime Text or VSCode as they help to run and test your code on multiple test-cases easily. However, in the lab you will be using command line interface.
- Try to use functions as much as possible in your code. Functions increase reusability and the pass-by-value feature provides a significant help sometimes. Modularizing your code also helps you to debug efficiently.
- You can implement your solutions in C or C++. Using C++ STL (Standard Template Library) is allowed.

Problem A. Weak Passwords

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

Main Idea is sorting a string in O(n) since we know it has only 'a'-'z'

You have created a password manager that is now being used by several people to store their passwords. Now, you want to add an additional functionality so that your users don't use similar passwords on multiple websites.

You do so by calculating the similarity score for each user as follows: given an array of strings containing all the passwords of a user, the similarity score is the number of pairs of passwords that are anagrams of each other. Implement a program to calculate this score. All the passwords are not necessarily of the same length. Also, the same password can be used by a user across multiple sites. Assume that each password only consists only lowercase alphabets.

Note:

Assume 1-based indexing.

The time complexity of your solution should be $O(nm + nlogn)$, where n is the number of passwords of the user and m is the maximum length of the passwords. Solution of higher time complexity will not be considered.

Input

The first line of input contains two integers, n ($1 \leq n \leq 10^4$) and m ($1 \leq m \leq 100$) — the number of passwords of the user and the maximum length of the passwords, respectively.

The second line of input contains n space separated strings $s_1, s_2, \ldots s_n$ ($1 \leq |s_i| \leq m$) — the passwords of the user on different websites.

Output

Print one line containing the similarity score of the user.

Example

standard input	standard output

`6`
`eat tea tan ate nat bat`
	`4`

Explanation

(s_1, s_2), (s_1, s_4), (s_2, s_4), (s_3, s_5) are the anagram pairs.

Problem B. Next Smallest Palindrome

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

DID NOT SOLVE

Given a numeric string A representing a large number, you need to find the next smallest palindrome strictly greater than this number.

Note:

The time complexity of your solution should be $O(n)$, where n is the length of string A. Solution of higher time complexity will not be considered.

Input

The first line of input contains n ($1 \leq n \leq 10^5$) — the length of the string A.

The second line of input contains the string A. It is guaranteed that A doesn't start with zeroes and only contains digits from 0-9.

Output

Print one line containing a numeric string denoting the next smallest palindrome strictly greater than A.

Examples

standard input	standard output
5 23545	23632

Problem C. Algebraic Manipulation

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

Revise This

You are given an array a of n integers and two additional integers, x and y.

Count the number of pairs of indices (i, j) such that $i < j$ and $a_j - a_i = yj - xi$.

Note:

Assume 0-based indexing.

The time complexity of your solution should be $O(nlogn)$. Solution of higher time complexity will not be considered.

Input

The first line of input contains n $(1 \leq n \leq 10^5)$ — the size of the array a.

The second line of input contains n space separated integers $a_0, a_1, \ldots, a_{n-1}$ $(1 \leq a_i \leq 10^5)$.

The third line of input contains two integers, x and y $(-10^4 \leq x, y \leq 10^4)$. x can be less than, equal to, or greater than y.

Output

Print one line containing the number of pairs of indices (i, j) such that $i < j$ and $a_j - a_i = yj - xi$. Print 0 if no such pair exists.

Examples

standard input	standard output

`6`
`3 5 1 4 6 6`
`2 -1`
	`2`

Explanation

$(0, 2)$ and $(1, 2)$ are the two pairs of indices that satisfy the given conditions.

Problem D. No Any Memory

Recall switch syntax and {}, initialization problems

Try to implement lower and upper bounds on your own, don't just rely on STL noob!

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

You have an array a of size n which is initialized with all 0s. You will be given q queries. Each query contains the query type and an integer k. There are 3 types of queries:

- Type 1: Flip the value of k-th element of a (if $a_k = 1$ then set $a_k = 0$ and vice-versa).
- Type 2: Print the smallest index x such that $x \geq k$ and $a_x = 1$. If no such index exists, print -1.
- Type 3: Print the largest index x such that $x \leq k$ and $a_x = 1$. If no such index exists, print -1.

Note:

Assume 1-based indexing.

The time complexity of your solution should be $O(qlogq)$, where q is the number of queries. Solution of higher time complexity will not be considered.

Input

The first line of input contains two integers, n $(1 \leq n \leq 10^9)$ and q $(1 \leq q \leq 10^5)$.

The following q lines contain two space separated integers, *type* $(1 \leq \textit{type} \leq 3)$ and k $(1 \leq k \leq n)$.

Output

For each query of type 2 or 3, print the required integer on a new line.

Example

standard input	standard output
`1000000000 7`	`200000000`
`1 4`	`5000`
`1 200000000`	`-1`
`1 5000`	
`2 5001`	
`1 200000000`	
`3 199999999`	
`2 10000`	

Problem E. Approved by Jilind

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

Easy question, just manual code writing

Jilind Main is trying to solve a problem and he needs your help (not really, he solved it instantly but we need a story). He has an array a of n integers and two other integers, m and k. For every subarray of a of length m, he wants to find the maximum of all the elements that appear exactly k times in the subarray.

Note:

A subarray is a contiguous part of the array.

Assume 1-based indexing.

The time complexity of your solution should be $O(nlogn)$. Solutions with higher time complexity will not be considered.

Input

The first line of input contains three integers n, m and k ($1 \leq k \leq m \leq n \leq 10^5$).

The second line of input contains n space separated integers $a_1, a_2, \ldots, a_n$ ($-10^9 \leq a_i \leq 10^9$).

Output

Print $n - m + 1$ lines. The i-th line should contain the maximum of all the elements that appear exactly k times in the subarray $[a_i, a_{i+1}, \ldots, a_{i+m-1}]$. If no such element exists in a subarray, print -1.

Examples

standard input	standard output

`6 4 1`
`3 3 3 4 4 2`
	`4`
	`-1`
	`3`

Explanation

In the given example,

- The first subarray is $[3, 3, 3, 4]$. 4 is the only element that appears 1 time in the subarray, so it is the maximum element as well.
- The second subarray is $[3, 3, 4, 4]$. No element appears 1 time in the subarray, so we print -1.
- The third subarray is $[3, 4, 4, 2]$. 3 and 2 are two elements that appear 1 time in the subarray, their maximum is 3.

Problem F. Bad XOR

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

Revise This

You are given an array of non-negative integers of size n. A subarray of the array is said to be bad if the *xor* of all its elements is 0. A subarray is defined as a contiguous part of an array. Find the total number of bad subarrays in the given array.

The time complexity of your solution should be $O(n * log(n))$. Any solution with a higher time complexity will not be considered.

Input

The first line consists of a single integer, n ($1 \leq n \leq 10^6$) — the number of elements in the array.

The second line of input contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^6$)

Output

Print a single line - the number of bad subarrays in the array.

Examples

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

Note

In the first example, the subarrays are (the first and last index of the sub array with 0-based indexing) - (0,2),(1,4) and (2,5)

In the second example the subarrays are - (0,1), (1,2), (3,4), (4,5), (5,6), (3,6), (1,4) and (1,6)

Problem G. Triangles

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

DID NOT SOLVE

You are given n points in the 2-D coordinate plane. Find the number of different triangles with non-zero area that can be formed with these points as the vertices.

The time complexity of your solution should be $O(n^2 * log(n))$. Any solution with a higher time complexity will not be considered.

Input

The first line consists of a single integer n $(1 \leq n \leq 1000)$ - the number of points in the plane

The next n lines consists of two space separated integers - x_i and y_i $(-10^9 \leq x_i, y_i \leq 10^9)$ denoting the x and y coordinates of the i^{th} point

Output

Print a single integer - the number of triangles that can be formed. Print 0 if no triangles can be formed.

Examples

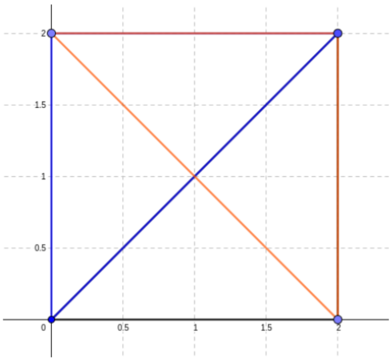
standard input	standard output

4
0 0
0 2
2 0
2 2
	4

7
0 0
0 1
0 2
1 0
2 0
1 1
2 2
	31

Note

In the first example the triangles will be formed by the points {(0,0),(0,2),(2,0)}, {(2,2),(0,2),(2,0)}, {(0,0),(0,2),(2,2)} and {(0,0),(2,2),(2,0)}



Problem H. LFI

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

You are the Lead Forensics Investigator (LFI) in a murder case. DNA has been found at the crime scene. The DNA sequence is composed of a series of nucleotides represented by the letters - A, G, C, and T. Finding repeated sequences in the DNA is required for DNA analysis. In the DNA sequence found at the crime scene, find all the sequences (substrings) of size 10 that occur more than once.

A sequence is said to be repeated if it occurs more than once in the string.

The time complexity of your solution should be $O(n * log(n))$ where n is the size of the input sequence

Input

The input consists of a single string, s $(1 \leq |s| \leq 10^4)$ - the DNA sequence found at the crime scene.

Output

In the first line print the number of sequences of size 10 that occur more than once. Print all sequences that occur more than once. Print 0 if no such sequence exists.

Examples

standard input	standard output

`GCTAAAAACCCCCAAAAACCCCCCAAAAAGGCTGTAGTT`	`2`
	`AAAAACCCCC`
	`CCCCCAAAAA`

`AAAAAAAAAAAAACGAT`	`1`
	`AAAAAAAAAA`

Problem I. Second Chance

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

You are conducting a tournament where so far n matches have been conducted. There are k players in the tournament and each player is represented by a unique integer. The result of each match is given by a 2 tuple ($winner_i$, $loser_i$) where $winner_i$ and $loser_i$ are the winner and loser of the i^{th} match respectively.

For the final rounds, you decide to advance only those people who have lost at most one match. Find all the players who will be advanced to the final rounds.

The time complexity of your solution should be $O(n * log(n))$ Any solution with a higher time complexity will not be considered.

Input

The first line consists of two integers, n and k $(1 \leq n, k \leq 10^5)$— the number of matches and the number of players respectively.

The next n lines consist of a pair of integers w_i and l_i $(1 \leq w_i, l_i \leq k)$, the winner and loser of the i^{th} match respectively.

It is guaranteed that every player plays at least one match.

Output

Print all the players who will advance to the final rounds in any order.

Examples

`standard input`	`standard output`

`10 10`
`1 3`
`2 3`
`3 6`
`5 6`
`5 7`
`4 5`
`4 8`
`4 9`
`10 4`
`10 9`
	`1 2 4 5 7 8 10`

`4 6`
`2 3`
`1 3`
`5 4`
`6 4`
	`1 2 5 6`

Note

In the first example

1. Players 1, 2, and 10 lost 0 matches
2. Players 4, 5, 7, and 8 lost 1 match
3. All the other players lost 2 or more matches.

Problem J. Too Bright

NICEEEEEEEEEEEEE

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

There are n light bulbs placed, some are lit and some are not. If a bulb is lit, it is denoted by 1 and if it is off, it is denoted by 0. You have to select a contiguous number of light bulbs such that at most k of them are lit. Find the number of ways you can select this contiguous sequence.

The time complexity of your solution should be $O(n * log(n))$. Any solution with a higher time complexity will not be considered.

Input

The first line consists of two integers n and k ($1 \leq n, k \leq 10^5$) — where n is the number of light bulbs and k is the maximum number of light bulbs lit in any selected continuous sequence

The second line consists of n space separated integers denoting if the light bulb is lit or not.

Output

Print a single integer — the number of contiguous sequences of light bulbs you can select such that at most k of them are lit.

Example

standard input	standard output

`5 2`
`1 0 0 1 1`
	`14`

`4 4`
`1 1 1 1`
	`10`

Note

In the first example, the contiguous sequences are (the first and last indices of the sub array with 0-based indexing) - (0,0), (0,1), (0,2), (0,3), (1,1), (1,2), (1,3), (1,4), (2,2), (2,3), (2,4), (3,3), (3,4) and (4,4). Note that a single element also denotes a contiguous sequence.