Codeforces Round #104 (Div. 1)

A. Lucky Conversion

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

*Petya loves lucky numbers very much. Everybody knows that lucky numbers are positive integers whose decimal record contains only the lucky digits **4** and **7**. For example, numbers **47**, **744**, **4** are lucky and **5**, **17**, **467** are not.*

Petya has two strings a and b of the same length n. The strings consist only of lucky digits. Petya can perform *operations* of two types:

- replace any one digit from string a by its opposite (i.e., replace 4 by 7 and 7 by 4);
- swap any pair of digits in string a.

Petya is interested in the minimum number of operations that are needed to make string a equal to string b. Help him with the task.

Input

The first and the second line contains strings a and b, correspondingly. Strings a and b have equal lengths and contain only lucky digits. The strings are not empty, their length does not exceed 10^5.

Output

Print on the single line the single number — the minimum number of operations needed to convert string a into string b.

Sample test(s)

input
47 74
output
1

input
774 744
output
1

input
777 444
output
3

Note

In the first sample it is enough simply to swap the first and the second digit.

In the second sample we should replace the second digit with its opposite.

In the third number we should replace all three digits with their opposites.

B. Lucky Number 2

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

*Petya loves lucky numbers very much. Everybody knows that lucky numbers are positive integers whose decimal record contains only the lucky digits **4** and **7**. For example, numbers **47**, **744**, **4** are lucky and **5**, **17**, **467** are not.*

Petya loves long lucky numbers very much. He is interested in the **minimum** lucky number d that meets some condition. Let $cnt(x)$ be the number of occurrences of number x in number d as a substring. For example, if $d = 747747$, then $cnt(4) = 2$, $cnt(7) = 4$, $cnt(47) = 2$, $cnt(74) = 2$. Petya wants the following condition to fulfil simultaneously: $cnt(4) = a_1$, $cnt(7) = a_2$, $cnt(47) = a_3$, $cnt(74) = a_4$. Petya is not interested in the occurrences of other numbers. Help him cope with this task.

Input

The single line contains four integers a_1, a_2, a_3 and a_4 ($1 \le a_1, a_2, a_3, a_4 \le 10^6$).

Output

On the single line print without leading zeroes the answer to the problem — the minimum lucky number d such, that $cnt(4) = a_1$, $cnt(7) = a_2$, $cnt(47) = a_3$, $cnt(74) = a_4$. If such number does not exist, print the single number "-1" (without the quotes).

Sample test(s)

input
2 2 1 1
output
4774

input
4 7 3 1
output
-1

C. Lucky Subsequence

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

*Petya loves lucky numbers very much. Everybody knows that lucky numbers are positive integers whose decimal record contains only the lucky digits **4** and **7**. For example, numbers **47**, **744**, **4** are lucky and **5**, **17**, **467** are not.*

Petya has sequence a consisting of n integers.

The subsequence of the sequence a is such subsequence that can be obtained from a by removing zero or more of its elements.

Two sequences are considered different if index sets of numbers included in them are different. That is, the values of the elements do not matter in the comparison of subsequences. In particular, any sequence of length n has exactly 2^n different subsequences (including an empty subsequence).

A subsequence is considered lucky if it has a length exactly k and does not contain two identical lucky numbers (unlucky numbers can be repeated any number of times).

Help Petya find the number of different lucky subsequences of the sequence a. As Petya's parents don't let him play with large numbers, you should print the result modulo prime number $1000000007\ (10^9+7)$.

Input

The first line contains two integers n and k $(1 \le k \le n \le 10^5)$. The next line contains n integers a_i $(1 \le a_i \le 10^9)$ — the sequence a.

Output

On the single line print the single number — the answer to the problem modulo prime number $1000000007\ (10^9+7)$.

Sample test(s)

input
3 2 10 10 10

output
3

input
4 2 4 4 7 7

output
4

Note

In the first sample all 3 subsequences of the needed length are considered lucky.

In the second sample there are 4 lucky subsequences. For them the sets of indexes equal (the indexation starts from 1): $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$ and $\{2, 4\}$.

D. Lucky Pair

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

*Petya loves lucky numbers very much. Everybody knows that lucky numbers are positive integers whose decimal record contains only the lucky digits **4** and **7**. For example, numbers **47**, **744**, **4** are lucky and **5**, **17**, **467** are not.*

Petya has an array a of n integers. The numbers in the array are numbered starting from 1. Unfortunately, Petya has been misbehaving and so, his parents don't allow him play with arrays that have many lucky numbers. It is guaranteed that no more than 1000 elements in the array a are lucky numbers.

Petya needs to find the number of pairs of non-intersecting segments $[l_1;r_1]$ and $[l_2;r_2]$ ($1 \le l_1 \le r_1 < l_2 \le r_2 \le n$, all four numbers are integers) such that there's no such lucky number that occurs simultaneously in the subarray $a[l_1..r_1]$ and in the subarray $a[l_2..r_2]$. Help Petya count the number of such pairs.

Input

The first line contains an integer n ($2 \le n \le 10^5$) — the size of the array a. The second line contains n space-separated integers a_i ($1 \le a_i \le 10^9$) — array a. It is guaranteed that no more than 1000 elements in the array a are lucky numbers.

Output

On the single line print the only number — the answer to the problem.

Please do not use the `%lld` specificator to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specificator.

Sample test(s)

input
4 1 4 2 4
output
9

input
2 4 7
output
1

input
4 4 4 7 7
output
9

Note

The subarray $a[l..r]$ is an array that consists of elements $a_l, a_{l+1}, ..., a_r$.

In the first sample there are 9 possible pairs that satisfy the condition: $[1, 1]$ and $[2, 2]$, $[1, 1]$ and $[2, 3]$, $[1, 1]$ and $[2, 4]$, $[1, 1]$ and $[3, 3]$, $[1, 1]$ and $[3, 4]$, $[1, 1]$ and $[4, 4]$, $[1, 2]$ and $[3, 3]$, $[2, 2]$ and $[3, 3]$, $[3, 3]$ and $[4, 4]$.

In the second sample there is only one pair of segments — $[1;1]$ and $[2;2]$ and it satisfies the condition.

E. Lucky Queries

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

*Petya loves lucky numbers very much. Everybody knows that lucky numbers are positive integers whose decimal record contains only the lucky digits **4** and **7**. For example, numbers **47**, **744**, **4** are lucky and **5**, **17**, **467** are not.*

Petya brought home string s with the length of n. The string only consists of lucky digits. The digits are numbered from the left to the right starting with 1. Now Petya should execute m queries of the following form:

- *switch* l r — "switch" digits (i.e. replace them with their opposites) at all positions with indexes from l to r, inclusive: each digit 4 is replaced with 7 and each digit 7 is replaced with 4 $(1 \le l \le r \le n)$;
- *count* — find and print on the screen the length of the longest non-decreasing subsequence of string s.

Subsequence of a string s is a string that can be obtained from s by removing zero or more of its elements. A string is called non-decreasing if each successive digit is not less than the previous one.

Help Petya process the requests.

Input

The first line contains two integers n and m $(1 \le n \le 10^6, 1 \le m \le 3 \cdot 10^5)$ — the length of the string s and the number of queries correspondingly. The second line contains n lucky digits without spaces — Petya's initial string. Next m lines contain queries in the form described in the statement.

Output

For each query *count* print an answer on a single line.

Sample test(s)

input
2 3 47 count switch 1 2 count
output
2 1

input
3 5 747 count switch 1 1 count switch 1 3 count
output
2 3 2

Note

In the first sample the chronology of string s after some operations are fulfilled is as follows (the sought maximum subsequence is marked with **bold**):

1. **47**
2. 74
3. **7**4

In the second sample:

1. 7**47**
2. 447
3. **447**
4. 774
5. **77**4