Problem A

You are given an array a of length N with all distinct integers and an integer C. You can select any number x and color all the elements a_i such that $a_i \geq x$. You can perform this operation exactly once. After performing this operation, a subarray of length K must exist such that $K \geq C$ and the number of colored elements exceed the number of uncolored elements. Find the minimum number of elements that need to be colored for this condition.

Input

The input consists of two lines. The first line contains two integers N and C ($1 \leq N, C \leq 10^5$). The second line contains N distinct integers $a_1, a_2, \ldots, a_N$ ($-10^9 \leq a_i \leq 10^9$).

Output

Output a single integer, which is the minimum number of elements that need to be colored.

Problem B

You are given an integer N. An array A is good if every subarray of length B has at least 2 distinct elements and all a_i lie in the range 1 to C. Your task is to find the number of good arrays of length N.

Input

The input consists of three integers N, C, and B ($1 \leq N \leq 10^9, 1 \leq C \leq 10^6, 1 \leq B \leq \min(50, N)$).

Output

Output a single integer representing the number of good arrays of length N.

Problem C

You are given two integers L and R. You are also given a binary search code snippet as follows:

```
while (l < r) {
    mid = (l + r) / 2;
    if (i < mid) {
        hi = mid - 1;
    } else if (i > mid) {
        lo = mid + 1;
    }else{
        return mid;
    }
}
```

The given binary search code finds the target value `i` in the $[L, R]$ range, but this code has a bug. You need to find the number of elements in the range $[L, R]$ that can't be found using the given code.

Input

The input consists of two integers L and R ($1 \leq L \leq 10^9, 1 \leq R \leq min(L + 10^5, 10^9)$), representing the range of values to search for.

Output

Output the number of elements in the range $[L, R]$ that can't be found using the given code.

Problem D

You are given an array arr consisting of N integers. You are also given q queries. Each query has three integers: X, Y, and Z. For each query:

1. Find the largest contiguous subarray B starting from index X whose Y-th bit is set.
2. Update each of its elements B_j with $B_j \oplus Z$, where $\oplus$ denotes the bitwise XOR operator.

Your task is to print the total number of updates performed after q queries.

Input

The input consists of the following:

- The first line contains an integer N, the size of the array.
- The second line contains N space-separated integers, representing the elements of the array arr.
- The third line contains an integer q, the number of queries.
- Each of the next q lines contains three space-separated integers X, Y, and Z, representing a query.

Output

Print a single integer, which is the total number of updates performed after q queries.

Constraints

- $1 \leq N \leq 10^5$
- $0 \leq a_i \leq 10^9$
- $1 \leq X \leq N$
- $1 \leq Y \leq 30$
- $0 \leq Z \leq 10^9$
- $1 \leq q \leq 10^5$