

Problem H. Papanashi

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 1024 mebibytes

Papanashi are just the best dessert ever made, something you can't imagine until you taste it. They are traditional Romanian fried or boiled doughnuts made with soft cheese, served warm with sour cream and sweet berry (usually blueberry) jam on top — a rich, fluffy, and indulgent dessert.



You want to give everyone papanashi as a dessert on your birthday. There are quite a few types of papanashi. You compiled some information about the preferences of each of your guests. For every guest i , you know the set a_i of types of papanashi that they like, represented as a bitmask: a set $\{v_1, \dots, v_k\}$ with elements ≥ 0 is written as the number $2^{v_1} + \dots + 2^{v_k}$.

Let x_i be the set of papanashi types person i gets to eat (represented as a bitmask). There are some constraints that you need to ensure in order for everything to be perfect.

1. There is a set of papanashi types s (represented as a bitmask). Every type of papanashi in s should be served to at least one guest, and everything not in s should not be served. Formally, $x_1 \text{ OR } x_2 \text{ OR } \dots \text{ OR } x_n = s$.
2. There is a set of papanashi types t (represented as a bitmask). Every type of papanashi in t should be served to an odd number of guests (this request came to you in a dream, so you must follow it to make the dream come true). Formally, $x_1 \text{ XOR } x_2 \text{ XOR } \dots \text{ XOR } x_n = t$.
3. Every guest should be served at least all the papanashi types that they like. Formally, $a_i \subseteq x_i$.

Since it's your birthday soon, you want to delight yourself with a problem. How many ways are there to serve everyone while still following the rules above? Two ways are different if some guest i gets different sets x_i . It is alright if the answer contains scenarios where you serve no papanashi to someone (it is not alright for the party though). Since this number can be very large, you want to compute it modulo $10^9 + 7$.

Input

The first line contains three integers: n , s , and t ($1 \leq n \leq 10^5$, $0 \leq s, t \leq 2^{30} - 1$).

The second line contains n integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq s$).

Output

Print a single integer: how many ways are there to serve everyone, modulo $10^9 + 7$.

Examples

<i>standard input</i>	<i>standard output</i>
3 11 10 8 2 0	12
3 7 5 2 3 6	0
10 31 24 5 17 1 6 0 30 12 15 8 23	8388608