The Mighty Spell

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

In a secret camp far far away, the great prophet, ftiasch, is trying to build up a new spell that could solve any ICPC problem with a single submission.

For this great propose, he has carefully crafted a string of n runes, or simply speaking, enchanted symbols. There are m different types of runes which are associated with different elements. For example, fire runes, water runes, light runes, dark runes, C++ runes, loli runes etc. A spell can be forged out of the string by activate some(probably none) of the runes in the string and deactivate the others.

To solve all the ICPC problems in the world is no easy task, even with the help of magic! So ftiasch would like to estimate the might of the spell he is going to get.

Basically, any spell works by rebalancing the flow of elemental energy in the world, so if there exists some element that the spell does not contain an activated rune of that kind, the spell won't work at all and thus have might 0.

Even if a spell contains all types of runes, the might still varies and that's where the crafted string kicks in. Magical bonds will form between consecutive activated runes and a consecutive segment of l runes with generate $g(l)$ units of might and the might of the spell is the sum of the mights of all its consecutive segments. After conducting heavy lots of research involving the forbidden arts of machine learning, ftiasch finally comes up with the explicit formula that $g(x) = 2x^3 + 3x^2 + 3x + 3$.

You might have been wondering why ftiasch doesn't simply activates all the runes in the string? Simply because ftiasch can't do that. And more importantly, that will make this problem too easy. In fact, each rune will be activated independently with probability $\frac{1}{2}$, and you have to calculate the expected might of the spell.

Input

The first line contains two integers n, m ($1 \le n \le 2 \times 10^5, 1 \le m \le 50$). Which are the length of the string and the number of types of runes.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le m$) which describe the string, c_i is the type of the i-th rune.

Output

A single integer denotes $\mathrm{E} \times 2^n \bmod (10^9 + 7)$ where E is the expected might of the spell.

Examples

standard input	standard output
3 2 1 2 2	152
4 3 1 2 1 2	0
6 3 1 2 3 3 2 1	3627