```cpp
/*
    Name: Binary Equivalence Problem
    Copyright: TCS Codevita S9
    Author: Sounish Nath
    Date: 22-09-20 00:08
    Description: A Problem B from my TCS Codevita Contest
*/

#include<bits/stdc++.h>
using namespace std ;
#define fo(i, n) for(int i = 0; i < n; i++)
#define foo(i, k, n) for(int i = k; i < n; i++)
#define deb(x) {cout << #x << " " << x << endl ;}
#define IOS ios::sync_with_stdio(false), cin.tie(0), cout.tie(0) ;
#define print(nums) { for(auto &&x : nums) { cout << x << " " ; } cout << endl ; }
#define MOD = 1e9+7 ;
const int N = 1e6+7 ;

int main(int argc, char const *argv[]){
    IOS ;

    int n ;
    cin >> n ;
    vector<int> nums(n) ;
    fo(i, n) {
        cin >> nums[i] ;
    }
    int maxelem = *max_element(nums.begin(), nums.end()) ;
    auto to_binary = [&](int n) {
        string bin = "" ;
        while(n) {
            bin += to_string(n % 2) ;
            n = n >> 1 ;
        }
        reverse(bin.begin(), bin.end());
        return bin ;
    };
    string maxelemBin = to_binary(maxelem) ;
    int maxelemLen = maxelemBin.length() ;
    unordered_map<int, pair<int, int>> cache ; // { int, {0, 1} }
    fo(i, n) {
        int x = nums[i] ;
        string xbin = to_binary(x) ;
        cout << xbin << " -> " ;
        if( (int) xbin.length() < maxelemLen) { xbin.insert(xbin.begin(), (maxelemLen - (int) xbin.length()), '0') ; }
        cout << xbin << endl ;
        int xcount0 = count(xbin.begin(), xbin.end(), '0') ;
        int xcount1 = count(xbin.begin(), xbin.end(), '1') ;
```

```
49:         cache[x] = {xcount0, xcount1} ;
50:     }
51:     int ans = -1;
52:     fo(i, (1 << n)) {
53:         int sum0, sum1 ;
54:         sum0 = sum1 = 0 ;
55:         for(int j = 0 ; j < n; j++) {
56:             if(i & (1 << j)) {
57:                 sum0 += cache[nums[j]].first ;
58:                 sum1 += cache[nums[j]].second ;
59:             }
60:         }
61:         if(sum0 == sum1) { ans++ ; }
62:     }
63:     cout << ans << endl ;
64:     string _ans = to_binary(ans);
65:     _ans.insert(_ans.begin(), (maxelemLen - (int) _ans.length()), '0') ;
66:     cout << _ans << endl ;
67:
68:     return 0;
69: }
```