// C++ program to calculate sum of Bit-wise

// and sum of all subsets of an array

#include <bits/stdc++.h>

using namespace std;

#define BITS 32

int andSum(int arr[], int n)

{

    int ans = 0;

    // assuming representation of each element is

    // in 32 bit

    for (int i = 0; i < BITS; i++) {

        int countSetBits = 0;

        // iterating array element

        for (int j = 0; j < n; j++) {

            // Counting the set bit of array in

            // ith position

            if (arr[j] & (1 << i))

                countSetBits++;

        }

        // counting subset which produce sum when

        // particular bit position is set.

        int subset = (1 << countSetBits) - 1;

        // multiplying every position subset with 2^i

        // to count the sum.

        subset = (subset \* (1 << i));

        ans += subset;

    }

    return ans;

}

// Drivers code

int main()

{

    int arr[] = { 1, 2, 3};

    int size = sizeof(arr) / sizeof(arr[0]);

    cout << andSum(arr, size);

    return 0;

}