Problem D. Not a subset sum

Time limit: 1 second
Memory limit: 1024 megabytes

We are given an array $a_0, a_1, ..., a_{2^n-1}$ of length 2^n. For a string q of length n over the alphabet `0,1,?` (1-indexed), define the "generalized subset sum"$S(q)$ as the sum of a_j over all indices j that satisfy:

- For each $1 \le k \le n$, if $q_k = 0$ then the k-th bit of j is 0.
- For each $1 \le k \le n$, if $q_k = 1$ then the k-th bit of j is 1.
- If $q_k =$? there is no restriction on the k-th bit of j.

For example, when $n = 3$ and $s =$ `0?1`, the generalized subset sum is $S(q) = a_4 + a_6$ (binary 100 and 110). **Note that the first bit is the lowest bit.**

Your task is to compute the generalized subset sum for each string of length n over the alphabet `0,1,?`. Because the total output can be large, you only need to output the **bitwise XOR** of all these sums.

Input

The first line of the input contains an integer n ($1 \le n \le 16$).

The second line of the input contains 2^n integers $a_0, a_1, \cdots, a_{2^n-1}$ ($0 \le a_i \le 9$), contents in array a.

Output

Print an integer denoting the bitwise-XOR of all generalized subset sums.

Examples

standard input	standard output
1 3 5	14
2 2 6 8 8	0

Note

The following are query strings q and their corresponding generalized subset sums $S(q)$ in **sample 2**:

$q =$ `00`, $S(q) = a_0 = 2$

$q =$ `10`, $S(q) = a_1 = 6$

$q =$ `?0`, $S(q) = a_0 + a_1 = 2 + 6 = 8$

$q =$ `01`, $S(q) = a_2 = 8$

$q =$ `11`, $S(q) = a_3 = 8$

$q =$ `?1`, $S(q) = a_2 + a_3 = 16$

$q =$ `0?`, $S(q) = a_0 + a_2 = 10$

$q =$ `1?`, $S(q) = a_1 + a_3 = 14$

$q =$ `??`, $S(q) = a_0 + a_1 + a_2 + a_3 = 24$

Now it is easy to verify that the output is 0.