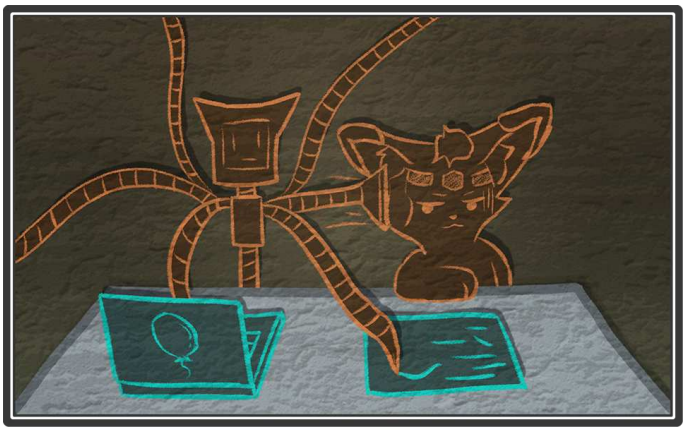
Problem F. Light Up the Hypercube

Time limit:	1 second
Memory limit:	1024 megabytes

Do androids dream of electric sheep? This topic, posed decades ago, stung with renewed urgency in future world, serving as a bitter reminder where biological minds had almost surrendered their thinking and creativity to artificial intelligence.

One afternoon in 2040, as Sulfox the fennec fox immersed himself in the pleasure of a simple two-dimensional puzzle game designed by Little Q (introduced in **Problem E. Light Up The Grid**), his solutions caught the attention of his custom AI system. The AI, despite having achieved nearly omnipotent capabilities in these years, failed to empathize with Sulfox's recreational purpose, misinterpreting his joy as a desire for greater challenges. In a display of its superior grasp of higher dimensions, the AI spontaneously provided an n-dimensional puzzle game that only it could truly comprehend.



Self-Important Custom AI System

The enhanced puzzle game involves an n-dimensional hypercube, where each of its edges extends along only one of the n dimensions, and each of its 2^n vertices is equipped with a light that can either be on or off. When a light is toggled, its state changes from off to on or from on to off.

Different from the simple puzzle game, there are 2^n kinds of toggle operations which can be performed at a certain cost, each numbered by an integer from 0 to $(2^n - 1)$. Specifically, for operation i $(i = 0, 1, \ldots, 2^n - 1)$:

- The cost of performing such operation equals a_i;
- Let $\overline{b_n b_{n-1} \ldots b_1}$ be the binary representation of i, where each $b_j \in \{0, 1\}$ $(j = 1, 2, \ldots, n)$ denotes the j-th bit of the binary representation from low to high, such operation allows selecting any valid k-face (subcube of dimensionality k) of the hypercube, where k is the number of $b_j = 1$ in the binary representation, each $b_j = 1$ constrains that the k-face selected must extend along the j-th dimension, and each $b_j = 0$ constrains that the k-face selected must have zero thickness in the j-th dimension. Then, toggle the lights at all vertices of the k-face selected.

For instance, operation 0 toggles the light at a single vertex (0-face) with cost a_0, operations $2^0, 2^1, \ldots, 2^{n-1}$ toggle lights at both endpoints of an edge (1-face) along their corresponding dimensions with cost $a_{2^0}, a_{2^1}, \ldots, a_{2^{n-1}}$ respectively, and operation $(2^n - 1)$ toggles all lights with cost a_{2^n-1}.

Sulfox had realized that the game had started, but the screen had been forbidden by the AI, blocking him from seeing the current state of any light. The only feedback he could receive is a special prompt sound triggered if, after an operation, all lights were on.

Therefore, he needed to find a sequence of operations, ensuring that regardless of the initial state of the lights (among all 2^{2^n} possibilities) when he started playing, he would always hear the prompt sound after some operations in the order of the sequence.

As a three-dimensional being, Sulfox felt impossible to visualize higher dimensions, yet believing the minimum total cost of such sequence of operations was still computable. "OK, AI, I want full manual control now," he said and set about solving, determined to swim free in the pool of cognition rather than be tethered by a lifejacket until forgetting how to stay afloat.

Please calculate the minimum total cost modulo 998 244 353 since it could be extremely large.

Input

The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains an integer n ($2 \le n \le 20$), indicating the dimensionality of the hypercube.

The second line contains 2^n integers $a_0, a_1, \ldots, a_{2^n-1}$ ($1 \le a_i \le 10^6$), indicating the costs of each operation.

It is guaranteed that the sum of 2^n over all test cases does not exceed 2^{20}.

Output

For each test case, output a line containing an integer, indicating the minimum total cost modulo 998 244 353.

Example

standard input	standard output
2 2 4 2 2 1 4 3 3 2 3 1 3 2 2 2 3 2 1 1 2 3 2	25 65666