Xor Mirror

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	64 megabytes

Long ago, in the land of Bitworld, there was a magic mirror. Whenever you looked into it, the index i of an array element would be reflected as $i \oplus k$ for some magic key k. The wizard who owned this mirror loved to play with sequences: sometimes he mirrored parts of the sequence using XOR, and sometimes he asked for weighted sums of intervals.

Now the wizard gives you T sequences and a list of operations for each sequence. Your task is to process them and report the results of his queries.

You are given an initial sequence $A_0, A_1, \ldots, A_{N-1}$ of length N. The value N is always a power of two and satisfies $N \leq 2^{18}$.

There are two types of operations:

- Operation type 1: given integers (l, r, k), for every $i \in [l, r)$ set

$$B_i = A_{i \oplus k},$$

  then assign $A_i = B_i$ for all $i \in [l, r)$.

- Operation type 2: given integers (l, r), output

$$\sum_{i=l}^{r-1} A_i.$$

Input

The first line contains a single integer T ($1 \leq T \leq 2 \times 10^5$) — the number of test cases.

Each test case is given as:

One line with two integers N ($1 \leq N \leq 2^{18}$, N is a power of two) and Q ($1 \leq Q \leq 2 \times 10^5$) — the length of the sequence and the number of operations.

One line with N integers $A_0, A_1, \ldots, A_{N-1}$ ($1 \leq A_i \leq 1048576$) — the initial sequence.

Each of the next Q lines describes one operation.

All operations are given on a left-closed, right-open interval $[l, r)$ with $0 \leq l < r \leq N$. The formats are:

`1 l r k` — apply operation type 1 on the interval $[l, r)$ with parameter k ($0 \leq k < N$).

`2 l r` — apply operation type 2 on the interval $[l, r)$.

It is guaranteed that $\sum N \leq 2^{18}$, $\sum Q \leq 2 \times 10^5$ over all test cases.

Output

For each operation of type 2, output the result on a separate line.

Example

standard input	standard output
1 8 8 7 3 8 1 4 6 4 1 2 2 7 2 5 7 1 3 5 3 1 5 6 2 2 7 8 2 3 7 1 2 8 5 2 5 8	23 10 1 13 22