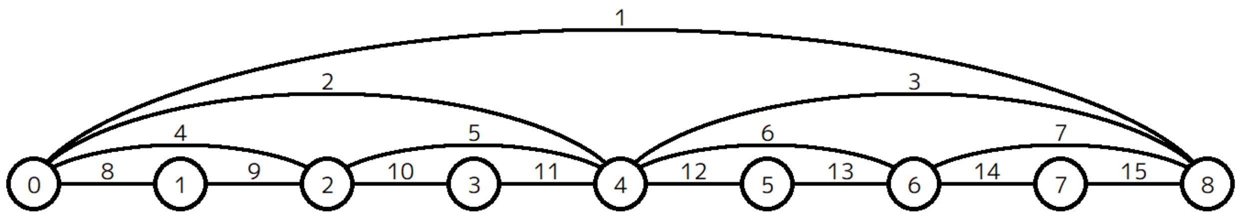
Segment Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 megabytes

You are given an undirected graph G with $2^N + 1$ vertices and $2^{N+1} - 1$ edges. The vertices are numbered $0, 1, \ldots, 2^N$, and the edges are numbered $1, 2, \ldots, 2^{N+1} - 1$.

Each edge in G belongs to one of $N + 1$ types, ranging from type 0 to type N. For type i $(0 \le i \le N)$, there are exactly 2^i edges, which are numbered $2^i + 0, 2^i + 1, \ldots, 2^i + (2^i - 1)$. The edge numbered $2^i + j$ $(0 \le j \le 2^i - 1)$ is an undirected edge of length C_{2^i+j} that connects vertex $j \times 2^{N-i}$ and vertex $(j + 1) \times 2^{N-i}$.

For example, when $N = 3$, G looks like the following graph:



You are given Q queries to process. There are two types of queries:

- `1 j x`: Change the length of edge j to x.
- `2 s t`: Find the shortest path length from vertex s to vertex t.

Input

The input is given in the following format. Note that vertex numbering starts from 0, while edge numbering starts from 1.

N
$C_1\ C_2\ \cdots\ C_{2^{N+1}-1}$
Q
query_1
query_2
$\vdots$
query_Q

Here, query_i represents the i-th query. Each query is given in one of the following formats:

`1` j x

`2` s t

- All input values are integers.
- $1 \le N \le 18$

- $1 \le C_j \le 10^7$ $(1 \le j \le 2^{N+1} - 1)$
- $1 \le Q \le 2 \times 10^5$
- In the query `1 j x`, $1 \le j \le 2^{N+1} - 1$ and $1 \le x \le 10^7$.
- In the query `2 s t`, $0 \le s < t \le 2^N$.
- There is at least one `2 s t` query.

Output

Let m be the number of queries of type `2 s t`. Output m lines, where the i-th line contains the answer to the i-th `2 s t` query.

Example

standard input	standard output
3 7 1 14 3 9 4 8 2 6 5 5 13 8 2 3 10 2 0 1 2 0 4 2 4 6 2 4 8 2 3 5 1 6 30 2 3 5 2 4 6 1 1 10000000 2 0 8	2 1 4 8 17 18 13 15

Note

- In the first query, using edge 8, the path $0 \to 1$ results in a total distance of 2.
- In the second query, using edge 2, the path $0 \to 4$ results in a total distance of 1.
- In the third query, using edge 6, the path $4 \to 6$ results in a total distance of 4.
- In the fourth query, using edges $2, 1$, the path $4 \to 0 \to 8$ results in a total distance of 8.
- In the fifth query, using edges $11, 6, 13$, the path $3 \to 4 \to 6 \to 5$ results in a total distance of 17.
- In the sixth query, the length of edge 6 is updated from 4 to 30.
- In the seventh query, using edges $11, 12$, the path $3 \to 4 \to 5$ results in a total distance of 18.
- In the eighth query, using edges $2, 1, 15, 14$, the path $4 \to 0 \to 8 \to 7 \to 6$ results in a total distance of 13.
- In the ninth query, the length of edge 1 is updated from 7 to 10000000.
- In the tenth query, using edges $2, 3$, the path $0 \to 4 \to 8$ results in a total distance of 15.