

Problem E

Dynamic Memory

Time limit: 2s
Memory limit: 1GB

A memory system consists of 2^k cells, numbered from 0 to $2^k - 1$. Each cell stores a single bit, either 0 or 1.

The system supports the operation $\text{STORE}(i, j, x)$, where $0 \leq i \leq k$, $0 \leq j < 2^{k-i}$ and $x \in \{0, 1\}$. This operation sets every cell in the interval $[j \cdot 2^i, (j+1) \cdot 2^i - 1]$, both ends included, to x . Its cost is w_i , regardless of j and x . Operations are performed one after another, so a later operation may overwrite values written by earlier operations.

You are also given a target configuration T consisting of 2^k bits. Initially, every bit of T is 0. There are q queries, each given by two integers L and R . A query flips every bit of T in the interval $[L, R]$, both ends included:

$$T[p] \leftarrow T[p] \oplus 1 \quad (L \leq p \leq R),$$

where $\oplus$ denotes bitwise XOR. The interval $[L, R]$ can be any contiguous interval: unlike the interval of a STORE operation, its length need not be a power of two and it need not satisfy any alignment restriction.

The queries change T cumulatively. After each query, find the minimum total cost of a sequence of STORE operations that turns a memory in which every cell contains 0 into the current configuration T . The operations considered for earlier queries play no role. The sequence may be empty, so the answer is 0 when every bit of T is 0.

Input

- The first line contains two integers k and q ($1 \leq k \leq 30$, $1 \leq q \leq 10^5$).
- The second line contains $k+1$ integers $w_0, w_1, \dots, w_k$ ($1 \leq w_i \leq 10^6$), where w_i is the cost of a STORE operation on an interval of length 2^i .
- Each of the next q lines contains two integers L and R ($0 \leq L \leq R < 2^k$), describing a query that flips the bits of T at all positions from L to R , inclusive.

Output

Print q lines. The t -th line must contain a single integer: the minimum total cost of obtaining the target configuration T after the t -th query.

The answers can exceed 10^{12} , which is beyond the range of a 32-bit integer type. Use a 64-bit integer type.

Sample Input	Sample Output
3 3 1 1 1 1 0 7 2 3 0 1	1 2 1
2 4 2 5 1 0 3 2 2 0 3 1 1	1 3 2 4

Notes

In the first example, there are $2^3 = 8$ memory cells and every **STORE** operation costs 1.

- After query 1, $T = [1, 1, 1, 1, 1, 1, 1, 1]$. The single operation **STORE**(3, 0, 1) sets all eight cells to 1. The minimum cost is 1.
- After query 2, $T = [1, 1, 0, 0, 1, 1, 1, 1]$. Apply **STORE**(3, 0, 1), followed by **STORE**(1, 1, 0). The minimum cost is 2.
- After query 3, $T = [0, 0, 0, 0, 1, 1, 1, 1]$. The single operation **STORE**(2, 1, 1) sets the last four cells to 1. The minimum cost is 1.

In the second example, $w_0 = 2$, $w_1 = 5$ and $w_2 = 1$. After query 2, $T = [1, 1, 0, 1]$. Applying **STORE**(2, 0, 1) and then **STORE**(0, 2, 0) costs $1 + 2 = 3$. If no operation were allowed to overwrite a cell written by an earlier operation, the cheapest way would be to set cells 0, 1 and 3 one by one, at a cost of 6.