

Maximum Profit

Input file: **standard input**
Output file: **standard output**
Time limit: 15 seconds
Memory limit: 1024 megabytes

Mr. L is the owner of a company with n employees and n positions. The i -th employee has a capability value of a_i , and the j -th position has a requirement value of b_j . The **base profit** generated when the i -th employee is assigned to the j -th position is: $a_i + b_j + (a_i \oplus b_j)$, where \oplus denotes the XOR operation in binary.

Additionally, Mr. L discovered that specific employee-position assignments yield **extra profits**. He provides m pieces of information in the form (x, y, w) , indicating that assigning the x -th employee to the y -th position results in an additional profit of w .

Given a parameter K , Mr. L wants to determine, for each $1 \leq k \leq K$, the **maximum total profit** (base profit + additional profit) achievable when exactly k employees are assigned to positions.

Note: Two employees cannot be assigned to the same position.

Input

The first line contains three integers n, m, K ($1 \leq n \leq 10^5, 0 \leq m \leq 5 \times 10^5, 1 \leq K \leq \min(300, n)$), denoting the number of employees/positions, the number of information entries, and the given parameter.

The second line contains n integers, where the i -th integer denotes a_i ($0 \leq a_i < 2^{12}$).

The third line contains n integers, where the i -th integer denotes b_i ($0 \leq b_i < 2^{12}$).

The next m lines each contain three integers x, y, w ($1 \leq x \leq n, 1 \leq y \leq n, 0 \leq w \leq 10^5$).

It is guaranteed that no two pieces of information share identical x and y values.

Output

Output a single line containing K integers, where the i -th integer denotes the answer when $k = i$.

Example

standard input	standard output
5 0 5	14 28 42 56 58
1 2 3 4 5	
5 4 3 2 1	