Sub array Sum2

Let A = {a0, a1, a2, a3, ..., an−1} be an array. We define a recursive operation Op on array A as follows

Op(A) = Op(two(A)) + Op(one(A)) + Op(zero(A)) if n > 1
= A otherwise

Here, zero(A) = {a0, a3, a6, ..} i.e. an array formed by elements whose indices are divisible by 3. Similarly, one(A) = {a1, a4, a7, a10, ...} and two(A) = {a2, a5, a8, a11..}. Also, + is the concatenation operation.

For example, if A = {0, 1, 2, 3, 4, 5}. Then Op(A) will be calculated as

Op(A) = Op({2, 5}) + Op({1, 4}) + Op({0, 3})
= Op({}) + Op({5}) + Op({2}) + Op({}) + Op({4}) + Op({1}) + Op({}) + Op({3}) + Op({0})
= {5, 2, 4, 1, 3, 0}

We define an query on an array B as taking the sum of all elements bk where i ≤ k ≤ j and l ≤ bk ≤ r.

We define C = {0, 1, 2, ..., n − 1}. So, you are given n and q queries and to have to perform q queries on B = Op(C)

Input

First line contains size n of array C. (n ≤ 10^15) -

Second line contains q, number of queries. (q ≤ 10^5) -

Next q lines contains four integers i, j, l, r. (0 ≤ i < n, i ≤ j < n, 0 ≤ l < n, l ≤ r < n)

Output

You have to output q integers modulo 10^9 + 7 corresponding to each query on a separate line.

Example

Input:

```
4
1
0 3 0 1
```

Output:

```
1
```