A Summatory (Extreme)

f(n) is defined as: f(n) = $1^k+2^k+3^k+...+n^k$, so it is the sum of the k-th power of all natural numbers up to n.

In this problem you are about to compute,

f(1) + f(2) + f(3) + ... + f(n)

Note: This is a harder version of the problem ASUMHARD, with larger constraints. Please read the constraints section carefully.

Input

The first line is an integer **T**, denoting the number of test cases. Then, **T** test cases follow.

For each test case, there are two integers **n** and **k** written in one line, separated by space.

Output

For each test case, output the result of the summatory function described above.

Since this number could be very large, output the answer modulo **1,234,567,891**.

Example

Input:
5
2 3
10 3
3 3
100 0
100 1

Output:
10
7942
46
5050
171700

Explanation

In case 1, **n** = 2, **k** = 3. f(1) = 1^3, f(2) = 1^3+2^3. **ans** = f(1) + f(2) = 10.

Constraints

Overall constraints

- $5 \leq \mathbf{T} \leq 500000$
- $1 \leq \mathbf{n} \leq 10^{18}$

- $0 \le \mathbf{k} \le 10000000$

More precise information (there are 6 test files):

Test #0: $\mathbf{T} = 500000$, $0 \le \mathbf{k} \le 100$

Test #1: $\mathbf{T} = 50000$, $0 \le \mathbf{k} \le 1000$

Test #2: $\mathbf{T} = 5000$, $0 \le \mathbf{k} \le 10000$

Test #3: $\mathbf{T} = 500$, $0 \le \mathbf{k} \le 100000$

Test #4: $\mathbf{T} = 50$, $0 \le \mathbf{k} \le 1000000$

Test #5: $\mathbf{T} = 5$, $0 \le \mathbf{k} \le 10000000$

It should be clear from the constraints that an $\mathbf{O(k^2)}$ solution **will not pass**. Inputs are generated uniformly randomly in the given ranges (with some manual worst case inputs). Time limit is set to 2x of my unoptimized C++ code.