Problem C
Faster Computation

You are given two arrays of integers $A_{1..N}$ and $B_{1..N}$. Your task is to compute the output of function $f(A, B)$ given by the following pseudocode:

```
function f(A[1..N], B[1..N]):
  ans = 0
  for i from 1 to N:
    for j from 1 to N:
      ans = ans + A[i] * A[j] * B[i] * B[j]
  return ans
```

As the output can be very large, you only need to print the output modulo $1\,000\,000\,007$.

The above function $f()$ has a time-complexity of $\Theta(N^2)$, and thus, it will run very slow for a large N, e.g., when $N = 50\,000$. It is your job to figure out how to speed-up the computation of function $f()$.

Input

Input begins with an integer T $(1 \leq T \leq 10)$ representing the number of cases.

Each case begins with an integer N $(1 \leq N \leq 50\,000)$. The second line contains N integers A_i $(1 \leq A_i \leq 10^6)$ representing the array A for $i = 1..N$, respectively. The third line contains N integers B_i $(1 \leq B_i \leq 10^6)$ representing the array B for $i = 1..N$, respectively.

Output

For each case, output in a line "`Case #X: Y`" (without quotes) where `X` is the case number (starts from 1) and `Y` is the output for the respective case.

Sample Input #1

```
3
4
1 2 3 4
1 2 3 4
5
10 2 3 7 4
1 2 5 3 2
2
100000 200000
300000 400000
```

Sample Output #1

```
Case #1: 900
Case #2: 3364
Case #3: 592900
```

Explanation for the sample input/output #1

For the 3^{rd} case,

- $i = 1, j = 1 : 100\,000 \times 100\,000 \times 300\,000 \times 300\,000 = 9 \times 10^{20}$
- $i = 1, j = 2 : 100\,000 \times 200\,000 \times 300\,000 \times 400\,000 = 24 \times 10^{20}$
- $i = 2, j = 1 : 200\,000 \times 100\,000 \times 400\,000 \times 300\,000 = 24 \times 10^{20}$
- $j = 2, j = 2 : 200\,000 \times 200\,000 \times 400\,000 \times 400\,000 = 64 \times 10^{20}$

Thus, the output is 121×10^{20}, modulo by $1\,000\,000\,007$ and you'll get $592\,900$.