# Digit Sum

**Time limit:** 1.0s **Memory limit:** 64M

His Imperial Majesty, the Dark Lord and Paramount Leader of Dmojistan knows that contest programmers really do not like to implement arbitrary precision integers. My Dark Master the Lord **quantum** knows this because a lot of contest problems simply ask for output in $\bmod 10^9+7$. Now, since My Dark Master the Lord **quantum** is a sadistic person, My Dark Master the Lord **quantum** would like to force you to implement such.

The problem is simple: given the integers $k$ ($0 \le k < 16\,384$) and $m$ ($0 \le m < 2^{64}$, except $k = 0$ in which case $m \in \{0, 2^{63}\}$), find the sum of digits of the integer $m \cdot 2^{k-63}$ in decimal. The resulting integer whose sum of digits you are to calculate is guaranteed to be an integer. Since you contest programmers are all so fond of $\bmod 10^9+7$, the output will be $\bmod 10^9+7$.

Since this is a pretty trivial task anyway, you can obviously fit your submission into 512 bytes. If not, well, a zero awaits you.

## Input Specification

The first line will be the number $N$ $(1 \le N \le 666)$, the number of cases to follow. Each case will be on its own line, containing the integers $k$ and $m$ for that case, in that order.

## Output Specification

For each case, output the sum of decimal digits of $m \cdot 2^{k-63}$ (which is an integer), in $\bmod 10^9+7$.

## Sample Input

```
4
4 9223372036854775808
10 10006998372017242112
19 11728112301486637056
63 13779283480589161437
```

## Sample Output

```
7
4
36
96
```

## Explanation

- $9\,223\,372\,036\,854\,775\,808 \cdot 2^{-63} \cdot 2^{4} = 1 \cdot 16 = 16, 1 + 6 = 7$.
- $10\,006\,998\,372\,017\,242\,112 \cdot 2^{-63} \cdot 2^{10} = 1111, 1 + 1 + 1 + 1 = 4$.
- $11\,728\,112\,301\,486\,637\,056 \cdot 2^{-63} \cdot 2^{19} = 666\,666, 6 \cdot 6 = 36$.
- $13\,779\,283\,480\,589\,161\,437 \cdot 2^{-63} \cdot 2^{63} = 13\,779\,283\,480\,589\,161\,437$, adding all digits result in 96.