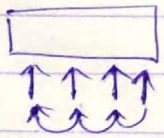
* Brute Force

According to the question we have to ignore those elements which contains the digit 9.

Initialize int count for counting N natural nos. & checking each number from 1 until count != N is containing digit 9 or not.

```
loop( count != N) {
    num
    num++
}
```



checking each no. wheather it contain digit 9 or not

* Optimized

Before getting optimized solution we have to understand number systems with different bases.

Decimal	Binary	Ternary	.....
0	0	0	
1	1	1	
2	10	2	
3	11	10	
4	100	11	
5	101	12	
6	110	20	
7	111	21	
8	1000	22	
9	1001	100	

From the above table we are getting that the base of Decimal is 10 so, ~~10~~ digits available in Decimal system is from 0 → 9 similarly

Base

Decimal (10) → {0 → 9}

Binary (2) → {0 → 1}

Ternary (3) → {0 → 2}

eg.

Decimal	Binary	Ternary
2	10	2
3	11	10
4	100	11

In Binary next element after 10 that doesn't contain digits from 2 → 9 is 11

also, " " " 11 " " " " "

" 100

So, we get that N^{th} ~~digit~~ number in decimal gives us N^{th} number in Binary representation that doesn't contain digit from 2 → 9

Ny

N^{th} number in decimal gives us N^{th} number in Ternary representation that doesn't contain digits from 3 → 9

⋮

" " " " " " " " " Base 9

" " " " digit = 9

So, just convert decimal N^{th} number to Base 9 representation to get N^{th} natural number that doesn't contain digit 9.

Decimal → Binary conversion

2 | 6	0	6 → 110
2 | 3	1	
2 | 1	1	
0		

Similarly do this for 9.

Summary.

For Binary Represnt^n allowed digits are only 0 → 1

" Ternary " " " " " " 0 → 2

" Base 9 " " " " " " 0 → 8