# Maths : Video - 33

Leetcode - 2081

Hard


codestorywithmik

CSwithMIK

codestorywithMIK


Try this channel to see "Life behind the Scenes + Tech News

# Motivation -

Growth is not always loud.

It happens quietly, day by day.

Keep learning, even when it feels slow.

MIK...

## 2081. Sum of k-Mirror Numbers

Hard | Topics | Companies | Hint

num

A **k-mirror number** is a **positive** integer **without leading zeros** that reads the same both forward and backward in base-10 **as well as** in base-k.

- For example, `9` is a 2-mirror number. The representation of `9` in base-10 and base-2 are `9` and `1001` respectively, which read the same both forward and backward.
- On the contrary, `4` is not a 2-mirror number. The representation of `4` in base-2 is `100`, which does not read the same both forward and backward.

Given the base `k` and the number `n`, return *the **sum** of the `n` **smallest** k-mirror numbers*.

Example :- K = 2, n = 5

Output :- 25

9 | 1001

K = 3, n = 7

Output :- 499

→ 1, 2, 4, 8, 121, 151, 212

→

| 3 | 8 | 2 |
|---|---|---|
| | 2 | |

22

K = 7, n = 17

output :- 20379000

# Thought Process

number base - 10 → Decimal numbers = 1, 2, 3, 4, 5, 6, 7 . . . . 99 . . .

number base - K →

| K=3 | 25 | 1 |
|---|---|---|
| 3 | 8 | 2 |
| | 2 | |

$(25)_{base=3}$ → 221

base - 2 → Binary

| 2 | 25 | 1 |
|---|---|---|

11001

Brute Force :- $n = 7, \quad K = 3$




Count = n

Count = 1
sum = 1 + 3

**steps:-**

① 1 to .... until we get n K-minu
↳ Palindro
↳ BacK → Pal. → K-mi

# Why trying brute force is bad in this problem ???

$n = 17, \quad K = 7$

7-minu

1, 2, 3, 4, 5, 6, 7, 8, ... 100, 121, .... 242 .. 292, ....

16561, ... 695859

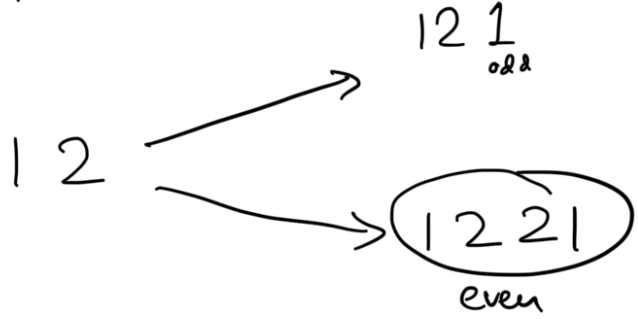
$6.9 * 10^6$

$n = 30$ large number.

# Optimal Approach

# We can't go and check all numbers #

decimal Palindromes




We know how to generate

Palindromes :-

→ Base K ⟶ Palin

# Generating Palindromes :-

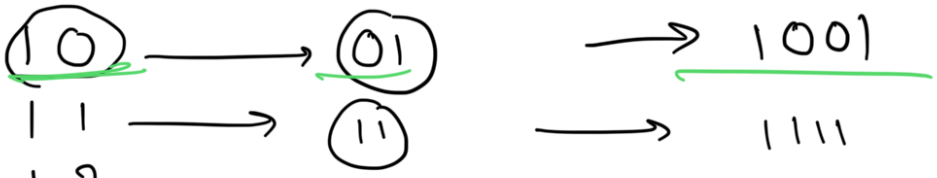
1, 2, 3, 4, . . . 9 ⟶ 1 length

11, 22, 33, 44, 55, 66, 77, 88, 99 → 2 length

101, 111, 121, 131, . . . . . ⟶ 3 length

4-length palindrome:- (Even length palindrome)


4/2



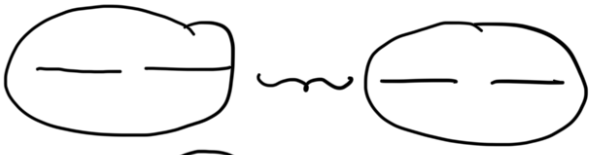

12
13
14
.
.
99

4 length → 4/2

L → Even

L/2 digits → numbers

## 5-length palindrome (odd length palindrome)



1 2 3 2 1

"24142"

100 → 10001

101 → 10101

102

103

104

⋮

858 → "85858"

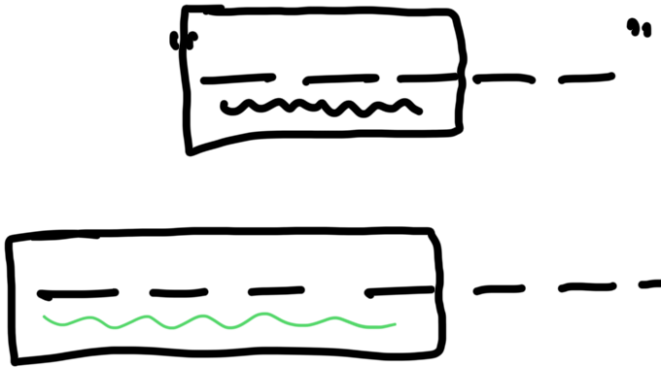
001

101

858

# L → odd

5, 7



$(L+1)/2$

$(5+1)/2 = 3$

$(7+1)/2 = 4$

1000

1234 → 1234321

~~4~~321

x

⋮

9999 → 99999991

L → even → L/2 digits generates nos.

↳ append with reverse.

L → odd → (L+1)/2 digits generate nos.

↳ apend with reverse except starting digit

# Generating Numbers :- of a given length.

```
if ( L % 2 == 1) //odd {
        half_length = (L+1)/2 ;
} else {
        half_length = L/2 ;
}
```

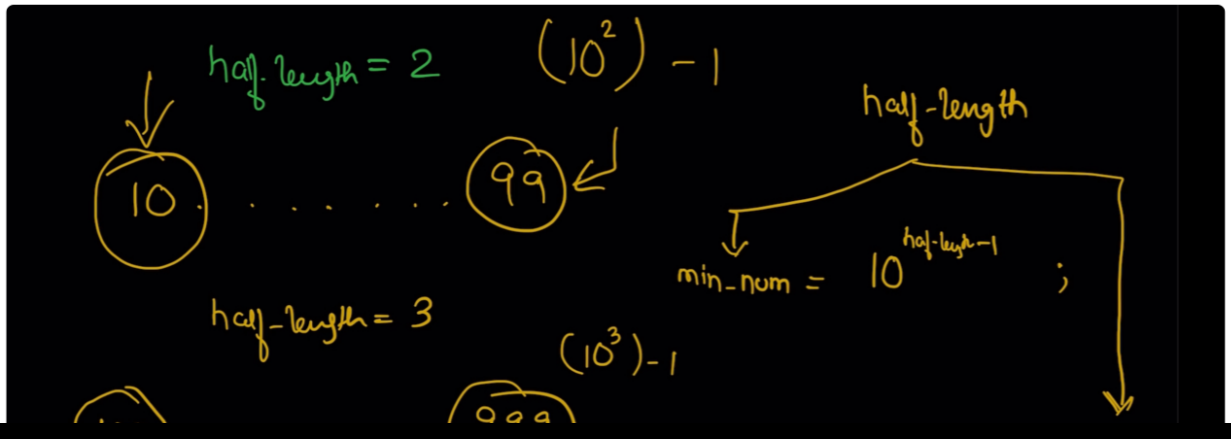
```
half_length = (L+1)/2 ;
```

L = 4

(4+1)/2 = 2

L = 5

(5+1)/2 = 6/2 = 3



100 . . . . . 111

max_num = (10^half_length) - 1;

```
long long min_num = pow(10, half_length - 1);
long long max_num = pow(10, half_length) - 1;
```

L = 6 . half_length = (6 + 1)/2 = 3

min_num = $10^{3-1}$ = 100;

max_num = $10^3 - 1$ = 999

```
for (num = min_num; num <= max_num; num++) {
```

"100"

length even ⇒ "100" "001"

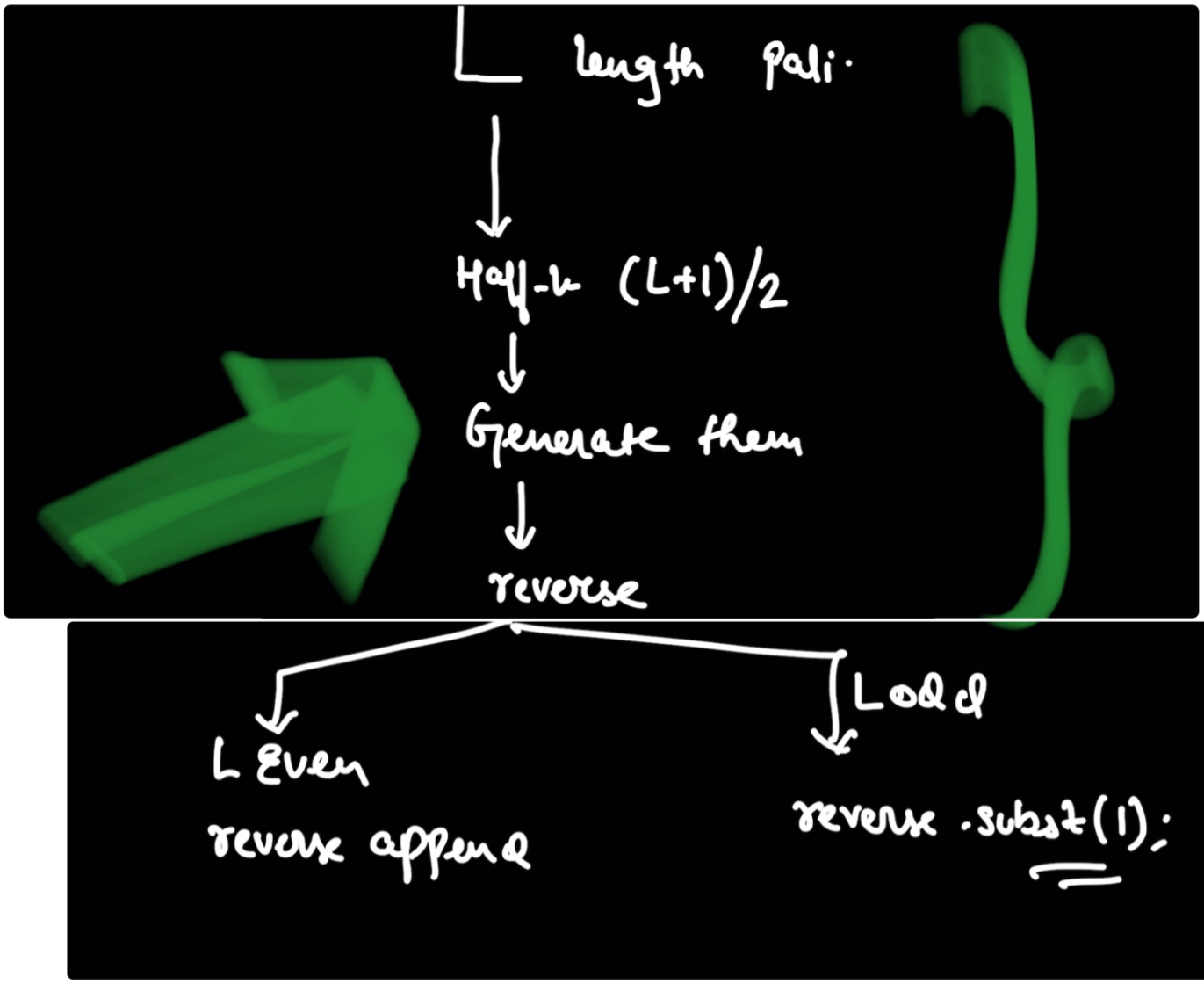
length odd ⇒

```
first_half = to_string(num); //1000
second_half = first_half;    : 1000
reverse(begin(second_half), end(second_half)); //"0001"

if (L → odd)
    first_half + second_half.substr(1).
```

e.g. "1000" & "001" → 1000 001

first half + secn half ;




## Time & Space :-

$K^{th}$ Mirror → N → digits = $(\log_{10} N)$

$$D = \log_{10} N$$

$$half = (D+1)/2$$

$$\begin{cases} min\_num & 10^{(D+1)/2 - 1} \\ max\_num & (10^{(D+1)/2}) - 1 \end{cases}$$

iterations $\rightarrow$ $10^{D/2}$

// generating Palin.

$\hookrightarrow$ $D$

base $k$. $\rightarrow$ $\log_k num$

$$\sim O(10^{D/2} * (D))$$

$$O(10^{D/2} * D), \quad D = \log_{10} N$$

Pol. $\longrightarrow$ $D$ $\quad \log_{10} N$

$\simeq$ $O(D)$