

LC 171. Excel Sheet Column Number

A → 1
B → 2
C → 3
⋮
Z → 26

AA → 27
AB → 28
⋮

It is similar to how Excel
names its Columns.

Implementation!

Str → "JAI"

Take 3 places

→ J is the 10th Alphabet.

So 1st gap can be filled in
10 ways.

$\frac{J}{\downarrow} - -$

10 ways.

Now, 2nd gap can be filled in
26 ways

$\frac{J}{\downarrow} \frac{J}{\downarrow} -$
10 26

Similarly 3rd gap can be filled in
26 ways.

$\frac{J}{\downarrow} \frac{J}{\downarrow} \frac{J}{\downarrow}$
10 26 26

So, strings that start with
'J' occurs after $(10 \times 26 \times 26)$
positions.

$\Rightarrow J \frac{A}{\downarrow} \rightarrow$ 26 ways.
1 way

\Rightarrow Strings that start with 'JA'
occurs after $(10 \times 26^2) + (1 \times 26)$
positions

$\Rightarrow J \frac{A}{\downarrow} \frac{I}{\downarrow} \rightarrow$ 9 ways.

\Rightarrow String JAI occurs after
 $(10 \times 26^2) + (1 \times 26) + (9)$ possibilities
from A(1)

So. JAIDHEER

$$\Rightarrow (10 \times 26^2) + (1 \times 26^1) + (9)$$

$$\Rightarrow (10 \times 26^2) + (1 \times 26^1) + (9 \times 26^0)$$

Similarly for JAIDHEER.

J	A	I	D	H	E	E	R
\downarrow							
10	1	9	4	8	5	5	18
\downarrow							
26^2	26^1	26^0	26^{-1}	26^{-2}	26^{-3}	26^{-4}	26^{-5}

$$\Rightarrow (10 \times 26^2) + (1 \times 26^1) + (9 \times 26^0) +$$

$$(4 \times 26^{-1}) + (8 \times 26^{-2}) + (5 \times 26^{-3}) +$$

$$(5 \times 26^{-4}) + (18 \times 26^{-5})$$

\Rightarrow It is similar to Number System where a Number of Base ' r ' is converted to Decimal ($r = 10$).

$$(553)_6 \Rightarrow ()_{10}$$

$$()_{10} = (5 \times 6^2) + (5 \times 6^1) + (3 \times 6^0)$$

As $r = 6$, the digits will be from 0 to 5

$$\Rightarrow ()_{10} = (5 \times 36) + (5 \times 6) + (3 \times 1)$$

$$= 180 + 30 + 3 = 213$$

$$\Rightarrow (553)_6 = (213)_{10}$$

Here, radix, $n = 26$

And digits are A to Z

We are converting $()_{26}$ to $()_{10}$

Means,

if column Number $()_{10}$ is given
we can convert it to string

$()_{26}$ using Repeated Division

It will be discussed after.

CODE

① Algorithm.

Given a string str

Let c-num = 1 (columnNumber)

let Base = 1 (26-Base).

Traverse the string from END
for Each char ch.

int ch_int = to_int(ch)

// Alphabet Number

// A → 1 ... Z → 26.

c_num += (ch_int * Base)

Base = Base * 26.

⇒ Initially Base = 1 (26^0)

⇒ for each iteration, Base will
multiplied by 26, i.e

$26^0, 26^1, 26^2, \dots$

```
long long titleToNumbers(string text){
    int text_len = text.length();
    long long base = 1;
    long long columnNumber = 0;

    for(int i=text_len-1;i>=0;i--){
        int ch_int = (text[i]-'A'+1);
        columnNumber += (ch_int * base);
        base *= 26;
    }

    return columnNumber;
}
```

Here Base is also taken as
long long as it is Exponential
growth of $\underline{2^6}$.

Having as Base Variable in an
Extra Space.

② without Base Variable

dit str = "JAI"

J	A	I
↓ 10	↓ 1	↓ 0
↓ 26 ²	↓ 26 ¹	↓ 26 ⁰

$$\Rightarrow (10 \times 26^2) + (1 \times 26^1) + (0 \times 26^0)$$

$$(10 \times 26^2) + (1 \times 26) + (9 \times 26^0)$$

$$\underline{((10 \times 26) + 1) \times 26 + 9}$$

H E L L O

H → 26^4
E → 26^3
L → 26^2
L → 26^1
O → 26^0
15

$$\Rightarrow (8 \times 26^4) + (5 \times 26^3) + (12 \times 26^2) + \\ (12 \times 26^1) + (15 \times 26^0)$$

$$\Rightarrow (8 \times 26^3) + (5 \times 26^2) + (12 \times 26) + 12$$

$$\times 26 + 15$$

$$\Rightarrow ((8 \times 26^2) + (5 \times 26) + 12) \times 26 + 12 \times 2 \\ + 15$$

$$\Rightarrow ((8 \times 26^2) + (5 \times 26) + 12) \times 26 + 12 + 15$$

$$\Rightarrow (((8 \times 26) + 5) \times 26 + 12) \times 26 + 12 + 15$$

$$\Rightarrow 8 \times 26 \Rightarrow ((8 \times 26) + 5) \times 26 + 12 + 12$$

$$\Rightarrow 8 \times 26 \Rightarrow (((8 \times 26) + 5) \times 26 + 12) \times 26 + 12$$

$$\Rightarrow (8 \times 26) + 5 \Rightarrow (((((8 \times 26) + 5) \times 26 + 12) \times 26 + 12) \times 26 + 12)$$

$$\Rightarrow ((8 \times 26) + 5) \times 26 \quad \underline{\times 26 + 12}$$

$$\Rightarrow ((8 \times 26) + 5) \times 26 + 12$$

$$\Rightarrow (((8 \times 26) + 5) \times 26 + 12) \times 26$$

⇒ If you analyze the Pattern
we are Traversing the String
from Start to End not from
End to Start

- ⇒ Initially 8 is taken (1st char)
- ⇒ Then it is multiplied with 26
- ⇒ To the result, 5 is added
(2nd char)
- ⇒ Then the result is multiplied
with 26
- ⇒ To the result, 12 is added. (3rd)
- ⇒ This Process is continued
upto last Character (15th)

```
long long titleToNumbers2(string text){  
    long long result = 0;  
    for(int ch_int:text)  
        result = (result * 26) + (ch_int - 64);  
    return result;  
}
```

here.

```
for (int ch-int : t.out)
```

in this for - each loop ch-int
variable is taken as int rather
than char.

Now ch-int contains the ASCII
code of the traversed character

ASCII of	Position of
A → 65	A → 1
B → 66	B → 2
:	:
Z → 90	Z → 26

∴ There is an offset of 64.

Just subtract it.

* Converting Number to Title.

* This can be done using Repeated division

let Cnum be the Number.

let rem = (C-num % 26)

Now add char(rem) to the output string

But Wait

% 26 output will range from
0 to 25

But here A is 1 and Z is 26

and % 26 will be zero only if
C-num = 26 K.

So if % 26 == 0, add 'Z' to
the result

and reduce n to $(n/26) - 1$ instead
of $n/26$ as we are working
the range.

for $\gamma_{26}, 1$ to 25. We will
simply add the correspondingly
alphabet and reduce n by
26.

```
string numberToTitle(long long number){
    string res = "";
    while(number){
        int rem = (number % 26);
        number /= 26;
        if(rem)
            res = char(rem + 64) + res;
        else{
            res = 'Z' + res;
            --number;
        }
    }
    return res;
}
```