

Day - 54Strings Interview Questions - 4* Roman to Integer

I → 1, V → 5, X → 10, L → 50,
 C → 100, D → 500, M → 1000

⇒ I → 1, II → 2, III → 3, IV → -1 + 5 = 4

⇒ If the next number is greater than the current number is considered as negative.

Ex:

$$CD = -100 + 500 = 400$$

Code

```

string s; // given
int sum = 0, index = 0;
while (index < s.size() - 1) {
    if (num(s[index]) < num(s[index + 1]))
        sum -= num(s[index]);
    else
        sum += num(s[index]);
    index++;
}
sum += num(s[s.size() - 1]);
return sum;
    
```

```

int num(char c) {
    if (c == 'I')
        return 1;
    else if (c == 'V')
        return 5;
    else if (c == 'X')
        return 10;
    else if (c == 'L')
        return 50;
    else if (c == 'C')
        return 100;
    else if (c == 'D')
        return 500;
    else
        return 1000;
}

```

* Integer to Roman:

1. 1248

$1000 + 200 + 40 + 8$

M + CC + XL + VIII

MCCXLVIII

For the conversion from 1000 to M, we will specify it.
 Also, in the same way, we will define for all the no. as 1 to 9, 10 to 100 as 10, 20, 30, ..., 100 & 1000 to 3000.

= Then we will break the no. and convert it to Roman.

* Factorial of a Number

= When the last time, we solved this problem, we got a issue that was Integer overflow.
 = There is a limit of int to store any no., that is 32 bit.

$$\begin{array}{c}
 \boxed{+} \rightarrow 2^{31} - 1 \\
 \text{32 bit} \\
 = 2^3 = 8 \\
 \left. \begin{array}{r}
 000 \\
 001 \\
 010 \\
 011
 \end{array} \right\} \text{So, } +ve \rightarrow 2^2 \\
 \text{Here, } 4 - 1 = 3 \rightarrow \text{for } 1, 2, 3 \\
 \left. \begin{array}{r}
 100 \\
 101 \\
 110
 \end{array} \right\} \text{And } 1 \text{ for } 0 \text{ (zero)} \\
 \overline{1+0} \\
 \overline{1+1=3}
 \end{array}$$

= At the time of factorial, we will store the intermediate calculations in a string.
 = For ex: 8

Date _____

Page _____

$$\begin{aligned} S &= "1" & - 8 \\ &= "8" & - 7 \\ &= "S6" & - 6 \\ &= "336" & - 5 \\ &= "1680" & - 4 \\ &= "6720" & - 3 \\ &= "20160" & - 2 \\ &= "40320" & - 1 \end{aligned}$$

- = In this method, we have to more calculations at the time of multiply.
= we can also do this in simpler way.
= we will take a vector ans

$$\begin{aligned} \text{ans} &= [1] \rightarrow 8 \\ &\quad 1 \times 8 \quad \swarrow \quad \searrow \\ &\quad 8 \quad \quad \quad 7 \\ &= [8] \quad \quad \quad 6 \\ &\quad 8 \times 7 \quad \swarrow \quad \searrow \\ &\quad = S6 \quad \quad \quad 3 \end{aligned}$$

$$\text{ans} = [6 \ 5] \quad 2$$

= we will store the result
in by breaking the number in different
positions of digits in diff. positions.
we will be storing the result in reverse
order because when we will get a carry in
the end then we can't add it to the
array.

$$\boxed{S \ 6} \times \boxed{3 \ 6}$$

Ex:

$$\begin{array}{r} \times 3031 \\ \hline \boxed{3} \ \boxed{3} \ \boxed{6} \end{array}$$

→ we can't store 3 in this way.

Code

```
vector<int> ans(1, 1);
while (N > 1) {
    int carry = 0; res > size = ans.size();
    for (i=0; i < size; i++) {
        res = ans[i] * N + carry;
        carry = res / 10;
        ans[i] = res % 10;
    }
    while (carry) {
        ans.push_back(carry % 10);
        carry /= 10;
    }
    N--;
}
```

~~return ans;~~

reverse(ans.begin(), ans.end());
return ans;