4-5-2021 [7:30-8:30] Hannah.

Reverse Integer

Given a signed 32-bit integer x, return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range $[-2^{31}, 2^{31}-1]$, then return 0.

Two tasks to focus on:

① Reverse the Integer

② If num < 0:

sign = −

① Reverse Integer.

Algorithm:-

1) rem = num % 10 ← ①

2) sum = sum × 10 + rem ← ②

3) num = num // 10 ← ③

Let's consider,

num = 123 & we know that O/P = 321

To reverse num we need to have the last-most digit as the first.

Let's use ① Algo for the extracting of last digit.

$$\begin{array}{r} 12 \\ 10\overline{)123} \\ -120 \\ \hline \boxed{3} \end{array} \rightarrow \text{Reminder}$$

Here we have extracted the last digit.

Now, let's use ② Algo to shift the last digit as it should be the first digit while reversing digit.
Here,
sum = 0
sum = 0 × 10 + 3
sum = 0 + 3
sum = 3

Now we will use the ③ Algo to have our second digit at the right position.
num = num // 10

$$\begin{array}{rl} & 1 \\ 10\,\big) & 12 \rightarrow \text{quotient from the last division} \\ & -10 \\ & \boxed{2} \rightarrow \text{Reminder}\end{array}$$

Now we will jump back to ① Algo & just have a loop implemented.
Recently, sum = 3.
sum = sum × 10 + rem
sum = 3 × 10 + 2
sum = 30 + 2
sum = 32

Now, we will take the quotient from last division i.e. ① & we will divide it by ⑩ but 1 is not divisible by 10.
∴ 1 will be the remainder.

we will use the ② Algo.
sum = sum × 10 + rem
sum = 32 × 10 + 1
sum = 320 + 1
sum = 321

Now, if the number is less than 0 then it ultimately means that the number is negative.

$\therefore$ if num < 0 :
    sign = -1
    num = -1 × num

The last condition states that the number should be a 32-bit signed integer. It means that it should be between the range of -2^{31} & $2^{31}-1$.

[

Palindrome Number

Given an integer x, return true if x is palindrome integer. An integer is a palin-drome when it reads the same backward as forward. For example, 121 is palindrome while 123 is not.

Here the first solution is going to be quite straightforward. We are going to convert the integer to a string i.e. 133 - "133". Then we are simply going to reverse the string by using step slicing.
Lets consider the input integer stored into a variable 'x'. After the conversion of integer to string we will store the string into a variable 'y'. Now we will reverse the string by step slicing → y[-1::-1] Now we will just compare the origional string and the reversed version of string.

```
y == y[-1::-1]
```

Roman to Integer

Given a roman numeral, convert it to an integer.
Roman numerals are represented by 7 - different symbols.
I, V, X, L, C, D & M.

Symbol	value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

For an example, 2 is written as II in Roman Numeral, just 2 one's added together. 12 is written as XII, which is simply X + II. The number 27 is writte as XXVII, which is XX + V * II

Algorithm to be used :-

① Iterate throughout. Take 2 pointers
  curr which is at position i.
  next which is at i + 1 position.

② Check curr >= next. Keep adding to output if not, add (next - curr) to output.

Now, lets dissolve the algorithm with the help of an example.
Let's consider the roman numeral LXVI i.e. 64.
But how will we use the algorithm to find out the integer value?

0 1 2 3 → These are the indices
L X I V

Now we are going to loop through all the roman integers and we are going to keep track of the current iteration (i) & the next iteration (i+1).
Over here, first iteration is L which stands for 50, & the next one is X which stands for 10.
Now we will check for a condition that is the current iteration greater than or equal to the next iteration [curr >= next]
If yes, then we will adding the first iteration to the output. O/P = 50.
Now the next time the first iteration is gonna be X which stands for 10, & the next one is I which stands for 1. 10 is obviously greater than 1, therefore O/P = 50 + 10 = 60.
Now the last time the first iteration will be I which stands for 1, & next iteration will be V which stands for 5. In this case 1 is not greater than 5. so, according to the algo. 5 − 1 = 4. therefore,
O/P = 60 + 4 = 64