Palindrome number

Companies :

Given an integer `x`, return `true` if `x` is palindrome integer.

An integer is a palindrome when it reads the same backward as forward.

- For example, `121` is a palindrome while `123` is not.

Example 1:

```
Input: x = 121
Output: true
Explanation: 121 reads as 121 from left to right and from right to left.
```

Example 2:

```
Input: x = -121
Output: false
Explanation: From left to right, it reads -121. From right to left, it becomes
121-. Therefore it is not a palindrome.
```

Example 3:

```
Input: x = 10
Output: false
Explanation: Reads 01 from right to left. Therefore it is not a palindrome.
```

Constraints:

- $-2^{31} <= x <= 2^{31} - 1$

Code :

```
class Solution {
public:
    bool isPalindrome(int x)
    {
        //negative can't be palindrome
        if(x < 0 || (x != 0 && x % 10 == 0))
        {
            return false;
        }
        //reerse the integer
           int rev=0;
            while(x>rev)
            {
                //rem = x%10;
                //rev =rev*10+rem;
                //x/=10;
                rev = (rev * 10) + (x % 10);
                x/=10;

            }
        //if reverse and original is same then its palindrome otherwise not
```

```
        if(x==rev || rev/10==x)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
};
```

Approaches :

2

1. Using to_strinng function
2. Simply reverse the number and check if it matches to original or not

I used 2nd approach :

Step - 1 : check if number is negative not zero(test case) and modulo 10 returns 0(test case) then it returns false

Step - 2 : simply reverse it.

Step -3 : if original and rev is match then it is palindrome and if odd number divide by 10 returns original number then its is true otherwise false.