Palindrome Number

Determine whether an integer is a palindrome. Do this without extra space.

click to show spoilers.

Some hints:
Could negative integers be palindromes? (ie, -1)

If you are thinking of converting the integer to string, note the restriction of using extra space.

You could also try reversing an integer. However, if you have solved the problem "Reverse Integer", you know that the reversed integer might overflow. How would you handle such case?

There is a more generic way of solving this problem.

Solution 1

```
class Solution {
public:
    bool isPalindrome(int x) {
        if(x<0|| (x!=0 &&x%10==0)) return false;
        int sum=0;
        while(x>sum)
        {
            sum = sum*10+x%10;
            x = x/10;
        }
        return (x==sum)||(x==sum/10);
    }
};
```

written by gaurav5 original link here

Solution 2

compare half of the digits in x, so don't need to deal with overflow.

```
public boolean isPalindrome(int x) {
    if (x<0 || (x!=0 && x%10==0)) return false;
    int rev = 0;
    while (x>rev){
        rev = rev*10 + x%10;
        x = x/10;
    }
    return (x==rev || x==rev/10);
}
```

written by cbmbbz original link here

Solution 3

```
public boolean isPalindrome(int x) {

    if (x < 0) return false;

    int p = x;
    int q = 0;

    while (p >= 10){
        q *=10;
        q += p%10;
        p /=10;
    }

    return q == x / 10 && p == x % 10;
}
```

// so the reversed version of int is always 1 time short in the factor of 10s .

in case of Int16, check 63556 will finally check if (6553 == 6355 && 6 == 63556%10) so there will have no concerns about the overflow.

written by evlstyle original link here

From Leetcoder.