Palindrome number

Given an integer x, return true if x is a , and false otherwise.

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.

This is surprisingly easy in Python first thing we do is eliminate any negative numbers

```
if x < 0:
            return False
```

Next we convert the numbers into a string and use python slicing to reverse the string

```
else:
            x = str(x)
            if x == x[::-1]:
                return True
            else:
                return False
```

The slicing is a bit difficult to understand at first but once you understand it becomes handy in alot of projects you do to say the least and has helped me with my human language technology project for creating a regular expression parser.

But essentially it is taken into 3 parameters [start:stop:step] so when the first two parameters are empty the -1 means step backwards true the sequence so this if statement matches if x is equal to == x[::-1] or 121 is equal to 121 wich would return true. If you wish to test the code for yourself play with these value

```
if __name__ == '__main__':
    x = 121 #Is a Palindrome
    y = -121 #Not a palindrome
    s = Solution()
    print(s.isPalindrome(x))
    print(s.isPalindrome(y))
```

Happy Coding 😃