Given a positive integer x, find the greatest number smaller than x that is a *palindrome*.

A number is considered to be a *palindrome* if its string representation looks the same when written backwards as it does when written forwards. For example, 1991, 7 and 808 are all *palindromes*.

**Example**

* For x = 16, the output should be  
  prevPalindrome(x) = 11.

11 is the greatest *palindrome* that is smaller than 16.

* For x = 11, the output should be  
  prevPalindrome(x) = 9.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer x**

An integer.

*Constraints:*  
1 ≤ x ≤ 109.

* **[output] integer**

The greatest *palindrome* number that is smaller than x.

<https://codefights.com/challenge/PQCRmQjcY32TARpWh>

static int prevPalindrome(int x)

{

for (int i = x-1; i >= 0; i--)

{

string n = i.ToString();

char[] rev = n.ToCharArray();

Array.Reverse(rev);

if (n == new string(rev))

{

return int.Parse( n);

}

}

return -1;

}