Write an algorithm to determine if a number n is happy.

```
//Happy number
// Importing HashMap class
import java.util.HashMap;
class Solution {
 //isHappyNumber() will determine whether a number is happy or not

    public boolean isHappy(int n) {
        HashSet<Integer> set = new HashSet();
        set.add(n);
        int sum = n;
        while(sum!=1){
            sum = getDigitSqrSum(sum);
            if(set.contains(sum)){
                return false;
            } else {
                set.add(sum);
            }
        }
        return true;
    }
      //Calculates the sum of squares of digits
    private int getDigitSqrSum(int n){
        int sum = 0;
        while(n!=0){
            int d = n%10;
            sum+=(d*d);
            n=n/10;
        }
        return sum;
    }
}
```

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

```
public class PalindromeNumber {

    public boolean isPalindrome(int x) {
        // Base condition
// If x is a negative number it is not a palindrome
        // If x % 10 = 0, in order for it to be a palindrome the first
digit should also be 0

        if (x < 0) {
            return false;
        }
        // Store the number in a variable
        int number = x;
        // This will store the reverse of the number
        int reverse = 0;
        while (number > 0) {
            reverse = reverse * 10 + number % 10;
            number /= 10;
        }
        return x == reverse;
    }
}
```