Problem Set 3 - Return Functions

p1: palindrome.py

A palindrome is a word that is spelled the same backward and forward, like 'noon' and 'redivider'. Recursively, a word is a palindrome if the first and last letters are the same and the remaining middle is a palindrome.

The following are functions that take a string argument and return the first, last, and middle letters:

```python
def first(word):
    return word[0]
def last(word):
    return word[-1]
def middle(word):
    return word[1:-1]
```

We shall study these techniques later on in the course.

1. Type these functions into a script and test them out. What happens if you call `middle()` with a string with two letters? One letter? What about the empty string, which is written '' and contains no letters?

2. Write a recursive function called `is_palindrome()` that takes a string argument and returns `True` if it is a palindrome and `False` otherwise. Remember that you can use the built-in function `len()` to check the length of a string.

p2: power.py

A number, \(a\) is a power of \(b\) if it is divisible by \(b\) and \(a/b\) is a power of \(b\). Write a recursive function called `is_power()` that takes parameters \(a\) and \(b\) and return `True` if \(a\) is a power of \(b\). Note: you will have to think about the base case.

p3: gcd.py

The greatest common divisor (GCD) of \(a\) and \(b\) is the largest number that divides both of them with no remainder.

One way to find the GCD of two numbers is base on the observation that if \(r\) is the remainder when \(a\) is divisible by \(b\), then \(\text{gcd}(a, b) = \text{gcd}(b, r)\). As a base case, we can use \(\text{gcd}(a, 0) = a\).
Write a recursive function called `gcd()` that takes parameters `a` and `b` and returns their greatest common divisor.