Practice Recursive Problems

Here are example solutions to the practice recursive problems.

String Reverser

This function reads a string and returns it in reverse order. The only string methods you should use are removing the first character and appending a char to a string

```
def reverser(input):
    # this takes a string and returns it in reverse order

    # base case, string of length 1
    if len(input) == 1:
        return input

    # call on all but first letter, append that to the end
    return reverser(input[1:]) + input[0]
```

Exponent

Write a recursive function that takes two arguments, base and exponent. It then multiplies the base times itself exponent times using recursion

```
def exponent(base, power):
    # this returns base times itself power times

    #base case, power <= 0
    if power <= 0:
        return 1

    #call itself, reduce power, do multiplication
    return base * exponent(base, power-1)
```

Palindrome detector

A palindrome is a string that reads the same forward and backward. Write a recursive function that takes a string and returns true if it is a palindrome. The algorithm is to compare the first and last letters, if they match, keep going with the inner string. If they do not match, return false. If the string is empty or one character, return true.

```
def isPalindrome(input):
    # this returns true if input is a palindrome

    #base case, string of zero or 1 letters
    if len(input) <= 1:
        return True

    if input[0] != input[-1]:
        return False

    return isPalindrome(input[1:-1])
```

Substrings

This is a more complex problem, you are to take a string and return all substrings of it. For example

A -> A
AB -> AB, A, B
ABC -> ABC, AB, AC, BC, B, C

```
def substrings(input):
    # this generates all substrings of a given input

    temp = [input]

        # add all substrings starting with first value
        # add all substrings starting without first value
    if len(input) > 0:
        temp.extend(substrings(input[1:]))
        temp.extend(substrings(input[:-1]))

        #use set to remove duplicates
    return list(set(temp))
```