CIS 122

Recursion Homework
Assignment 2

● Four recursion themed problems
  ○ Triangle warmup
  ○ Puzzling Palindromes
  ○ Collatz Quandary
  ○ A Shifty Problem (part 2)
Part 0 - Triangle Warmup

• How many dots does it take to draw a triangle?
  ○ 1 dot on the first row
  ○ 2 dots on the second row
  ○ 3 dots on the third row
  ○ ...

1   3   6   10
Part 0 - Triangle Warmup

● How many dots does it take to draw a triangle?
  ○ 1 dot on the first row
  ○ 2 dots on the second row
  ○ 3 dots on the third row
  ○ ...

  1 3 6 10

● How many dots in a triangle of size 1?
  ○ 1
Part 0 - Triangle Warmup

● How many dots does it take to draw a triangle?
  ○ 1 dot on the first row
  ○ 2 dots on the second row
  ○ 3 dots on the third row
  ○ ...

1 3 6 10

● How many dots in a triangle of size 2?
  ○ 1 + 2 = 3
Part 0 - Triangle Warmup

● How many dots does it take to draw a triangle?
  ○ 1 dot on the first row
  ○ 2 dots on the second row
  ○ 3 dots on the third row
  ○ ...

● How many dots in a triangle of size 3?
  ○ $1 + 2 + 3 = 6$
Part 0 - Triangle Warmup

● How many dots does it take to draw a triangle?
  ○ 1 dot on the first row
  ○ 2 dots on the second row
  ○ 3 dots on the third row
  ○ ...

1  3  6  10

● How many dots in a triangle of size n?
  ○ 1 + 2 + 3 + ... + n
Part 0 - Triangle Warmup

- Write a function `triangle(n)`
  - Calculate the nth triangle number
  - $1 + 2 + 3 + \ldots + n$

- What's our base case?

- What's our recursive step?
Part 0 - Triangle Warmup

● Write a function `triangle(n)`
  ○ Calculate the nth triangle number
  ○ $1 + 2 + 3 + \ldots + n$

● What's our base case?
  ○ `triangle(0) = 0`
  ○ `triangle(1) = 1` if your prefer

● What's our recursive step?
Part 0 - Triangle Warmup

- Write a function `triangle(n)`
  - Calculate the nth triangle number
  - \(1 + 2 + 3 + \ldots + n\)

- What's our base case?
  - `triangle(0) = 0`
  - `triangle(1) = 1` if your prefer

- What's our recursive step?
  - `triangle(n) = n + triangle(n-1)`
Part 0 - Triangle Warmup

● Write a function \texttt{triangle(n)}
  ○ Calculate the nth triangle number
  ○ $1 + 2 + 3 + \ldots + n$

● What's our base case?
  ○ \texttt{triangle(0)} = 0
  ○ \texttt{triangle(1)} = 1 if your prefer

● What's our recursive step?
  ○ \texttt{triangle(n)} = n + \texttt{triangle(n-1)}

● Very similar to factorial...
Part 1 - Puzzling Palindromes

- A palindrome reads the same forwards and backwards
  - pop
  - madam
  - racecar

- Write a function `isPalindrome(word)`
  - Return `True` if word is a palindrome
  - Return `False` otherwise
Part 1 - Puzzling Palindromes

- How do we tell if a word is a palindrome?

RACECAR
Part 1 - Puzzling Palindromes

• How do we tell if a word is a palindrome?
  ○ First and last letters must match
Part 1 - Puzzling Palindromes

- How do we tell if a word is a palindrome?
  - First and last letters must match
  - Rest of word must be a palindrome

RACECAR
Part 1 - Puzzling Palindromes

- How do we tell if a word is a palindrome?
- Base Case
- Recursive Step
Part 1 - Puzzling Palindromes

- How do we tell if a word is a palindrome?
- Base Case
  - The empty string is a palindrome
  - If first and last letters don't match, NOT a palindrome
- Recursive Step
Part 1 - Puzzling Palindromes

● How do we tell if a word is a palindrome?

● Base Case
  ○ The empty string is a palindrome
  ○ If first and last letters don't match, NOT a palindrome

● Recursive Step
  ○ If first and last letters match...
  ○ The rest of the word must be a palindrome
Part 2 - Collatz Quandary

- HOTPO - Half Or Triple Plus One
  - If number is even, divide it by two
  - If number is odd, multiply by three and add one

- Collatz Conjecture
  - Pick any number
  - Repeatedly apply HOTPO
  - Eventually, it will reach 1

- You don't need to prove the Collatz Conjecture
  - It's still an open problem
Part 2 - Collatz Quandary

- Write a function `collatz(n)`
  - How many steps does it take to get from `n` to 1?
Write a function `collatz(n)`
  ○ How many steps does it take to get from n to 1?

We know what the next number in the sequence is
  ○ If n is even, it's n/2
  ○ If n is odd, its 3*n+1

What if we knew how long it took that number to get to 1?
  ○ Clearly, it takes n one step more!
Part 2 - Collatz Quandary

- Write a function `collatz(n)`
  - How many steps does it take to get from n to 1?
- Base Case
- Recursive Step
Part 2 - Collatz Quandary

• Write a function `collatz(n)`
  ○ How many steps does it take to get from `n` to 1?

• Base Case
  ○ `collatz(1) = 0`
  ○ It's already there!

• Recursive Step
Part 2 - Collatz Quandary

- Write a function `collatz(n)`
  - How many steps does it take to get from n to 1?

- Base Case
  - `collatz(1) = 0`
  - It's already there!

- Recursive Step
  - Find how many steps the number after n takes
  - n takes one step more
Part 3 - A Shifty Problem (part 2)

- Last week, you wrote a single character shifter
  - Takes a character, and a number
  - Shifts character forward by that number

- Now, we'll use that work to encipher entire strings
  - encipher(text, num) - Shift entire text forward by num
  - decipher(text, num) - Shift entire text back by num

- Before you do any work, copy over caesarShift
  - Make sure it works
  - You can use the solution function instead...
Part 3 - A Shifty Problem (part 2)

- How can we encipher an entire string?
  - We know how to encipher single characters
  - Let's encipher our string in pieces
Part 3 - A Shifty Problem (part 2)

- How can we encipher an entire string?
  - We know how to encipher single characters
  - Let's encipher our string in pieces

AAAAAAA
Part 3 - A Shifty Problem (part 2)

- How can we encipher an entire string?
  - We know how to encipher single characters
  - Let's encipher our string in pieces

AAAAAAA
Part 3 - A Shifty Problem (part 2)

- How can we encipher an entire string?
  - We know how to encipher single characters
  - Let's encipher our string in pieces

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A

A
Part 3 - A Shifty Problem (part 2)

- How can we encipher an entire string?
  - We know how to encipher single characters
  - Let's encipher our string in pieces

- Base Case

- Recursive Step
Part 3 - A Shifty Problem (part 2)

- How can we encipher an entire string?
  - We know how to encipher single characters
  - Let's encipher our string in pieces

- Base Case
  - The empty string can be trivially encoded

- Recursive Step
How can we encipher an entire string?
  ○ We know how to encipher single characters
  ○ Let's encipher our string in pieces

Base Case
  ○ The empty string can be trivially encoded

Recursive Step
  ○ Shift the first character
  ○ Encode the rest of the string
Part 3 - A Shifty Problem (part 2)

● How do we decode?
  ○ Easier than it sounds

● Recall that shifting forward by 26 means not shifting at all
  ○ If we've already encoded our string
  ○ We can "encode" it even more
  ○ Wrap all the way around