Recursion I

Factorials are Factorials Times Factorials

Factorials are defined as the product of an integer with all integers below it. For example, factorial(5) = 5 x 4 x 3 x 2 x 1. Fill in the code below to recursively compute a factorial.

```
factorial(n):
    if n = 1 :
        report 1
    else:
        report n x factorial (n-1)
```

PalindromeemordnilaP

(a) A palindrome is a word that is spelled the same way forwards and backwards. In other words, the first letter must equal the last letter, the second letter must equal the second to last letter ... etc.

Using the above information, fill in the recursive palindrome function. You have access to these two functions:

```
all-but-first-of( word )
```

reports *word* with the first letter gone

```
all-but-last-of( word )
```

reports *word* with the last letter gone

```
is (word) a palindrome?:
    if length(word) < 2 :
        report true
    else:
        if letter (1) of (word) = letter (length(word)) of (word) :
            report is (all-but-last-of (all-but-first-of( word ))) a palondrome?
        else:
            report false
```

(b) Fill in the progression of calls to: is (racecar) a palindrome?

is (racecar) a palindrome?

is (aceca) a palindrome?

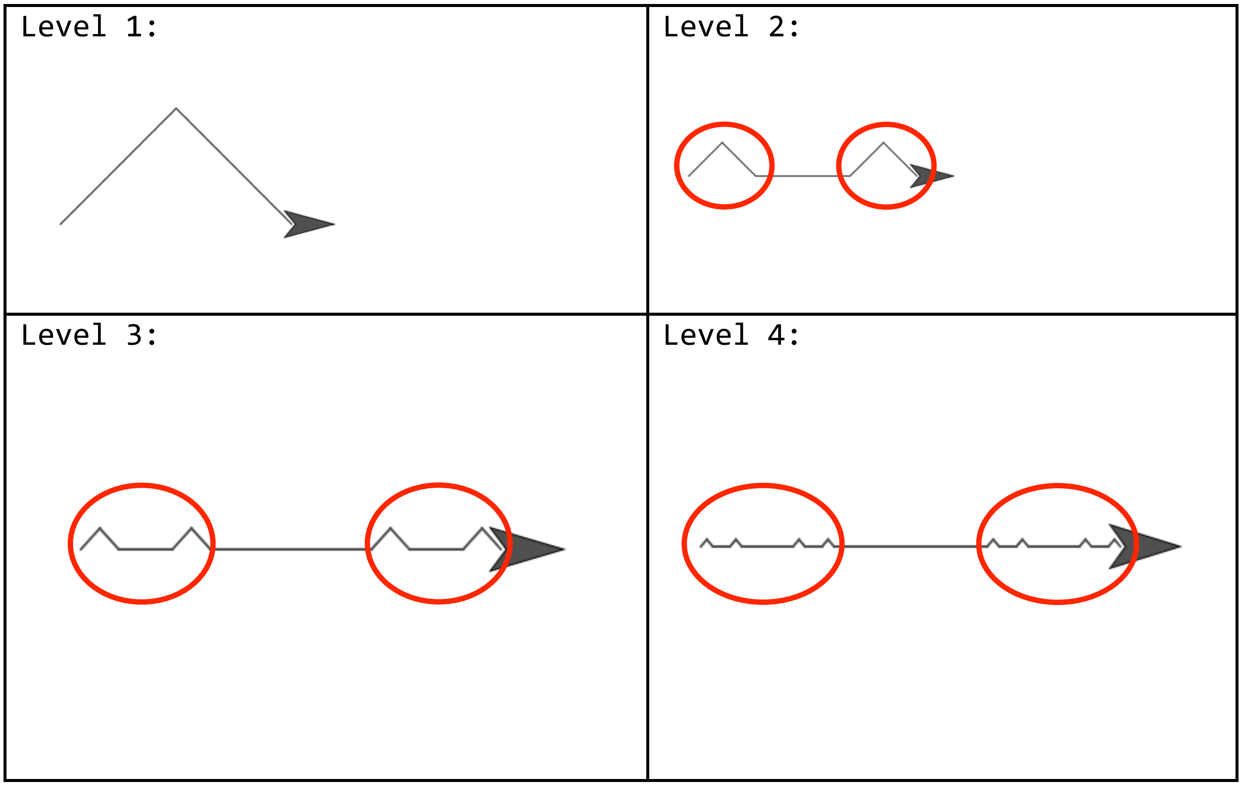
is (cec) a palindrome?

is (e) a palindrome?

true

Where are These Cats Coming From?!

In the following exercise, we will address how to construct and how to think about fractals recursively. You may assume that the sprite starts off at the leftmost part of each level, facing right.



1) Which level corresponds to the base case? level 1

2) For our base case, in what direction does our sprite start and end in?

It starts and ends facing right.

3) In each level, circle each instance of the previous level. Each of these instances refers to one recursive call.

4) What does the sprite do between each of the recursive calls?

Move size steps.