Internet and Big Data

Recursion

Credit: Melanie

Theory and Understanding

Question 1: Matching

- base case
- recursion
- recursive call
- recursive definition
- infinite recursion

1. The statement that calls an already executing function.
2. A definition which defines something in terms of itself. To be useful it must include base cases which are not recursive.
3. A branch of the conditional statement in a recursive function that does not give rise to further recursive calls.
4. A function that calls itself recursively without ever reaching the base case.
5. The process of calling the function that is already executing.

Question 2: True or False

```
def mystery(numList):
    value = 0
    for num in numList:
        value = value + num
    return value
print(mystery([1,3,5,7,9]))
```

True or false? The code above can be rewritten recursively.

Coding

Q3. Reverse a List

Write a recursive function called **reverseList(lst)** that reverses a list.

https://runestone.academy/runestone/books/published/thinkcspy/IntroRecursion/ProgrammingExercises.html

Q4. Vowel Counter

Write a function called **count_vowels(st)** that returns the number of vowels in the string **st**, using recursion.

Q5. Palindrome Checker

Write a function **is_palindrome(word)** that uses **recursion** to check whether **word** is a palindrome. It should return True if the word is a palindrome and False if it is not.

E.g.

- A and ABA should return True
- AB and ABC should return False

Extra Practice

Question 1

Given a list of **integers** and a **search** term, write a **search** function that will **return a list** containing **all the indices** where the search term can be found. If it cannot be found, return an empty list. Your solution should use the **append** function. You must write at least 3 test cases.

```
def linear_search_multiple(input_nums,search_term):
```

Question 2

Write a function **is_palindrome(word)** that uses a **loop** to check whether **word** is a palindrome. It should return True if the word is a palindrome and False if it is not.

Question 3

Write a **recursive binary search** function that takes as input a **sorted** list of integers and an integer search term, and returns a **boolean** value indicating whether the value is in the list or not.

```
def recursive_binary_search(input_nums,search_term):
```

Hint:
- You can call the function recursively depending on the situation