Count Vowels in a String

In this lesson, we will learn how to find the number of vowels in a string using recursion.

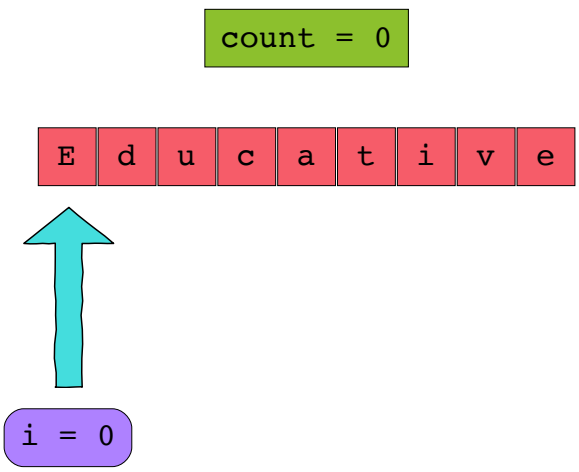
We'll cover the following

Problem Statement

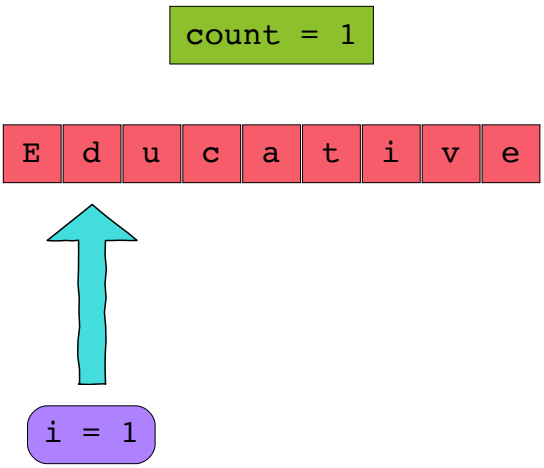
Imagine that we have to find the number of **vowels** in a given string. We know that the **English alphabet** contains 5 vowels: a, e, i, o, u. Let's solve this problem using both iterative and recursive methods.

Iterative Method

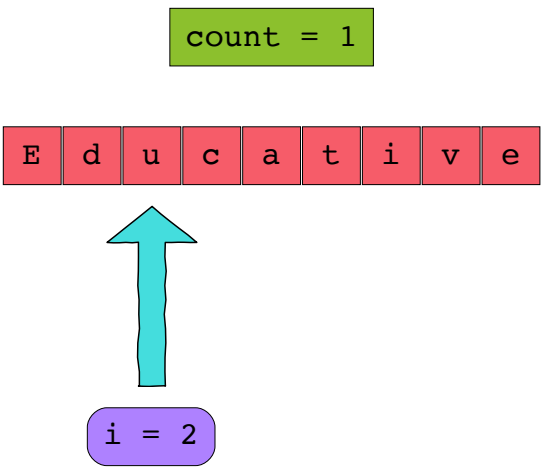
To solve this problem we have to traverse the entire string so that we can initiate a simple `for` loop.



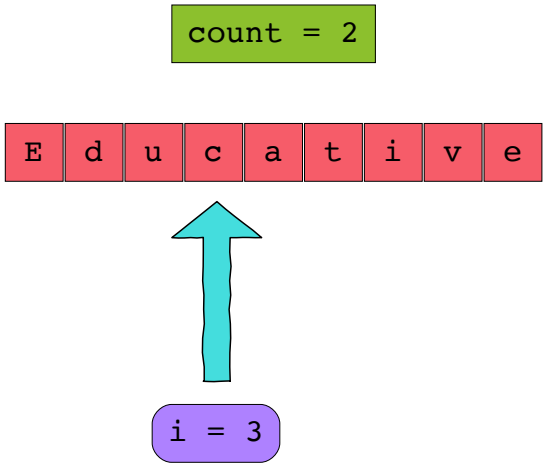
Initiate a for loop



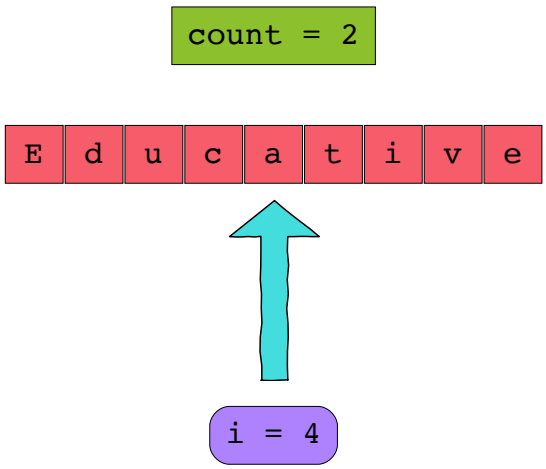
Iteration 1



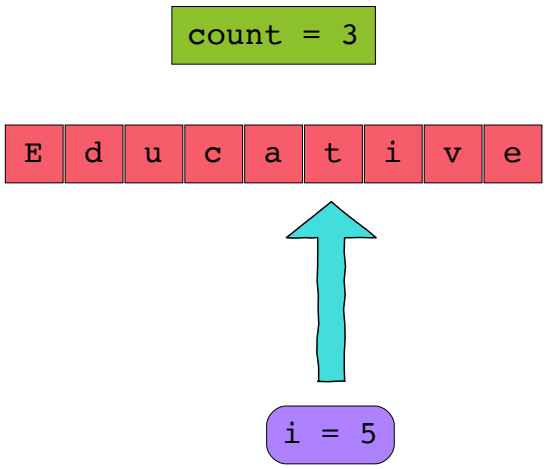
Iteration 2



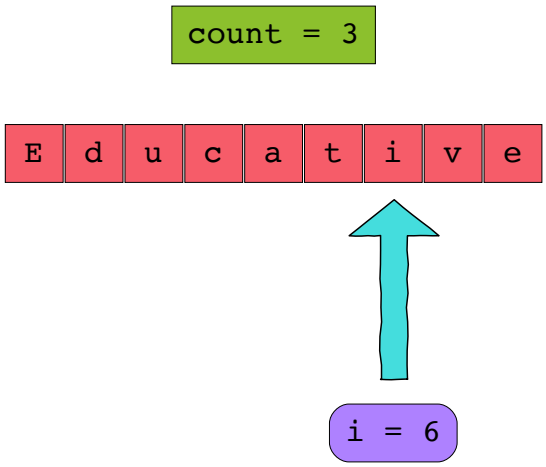
Iteration 3



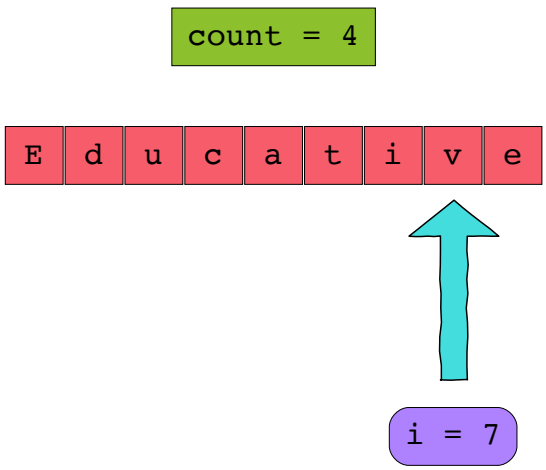
Iteration 4



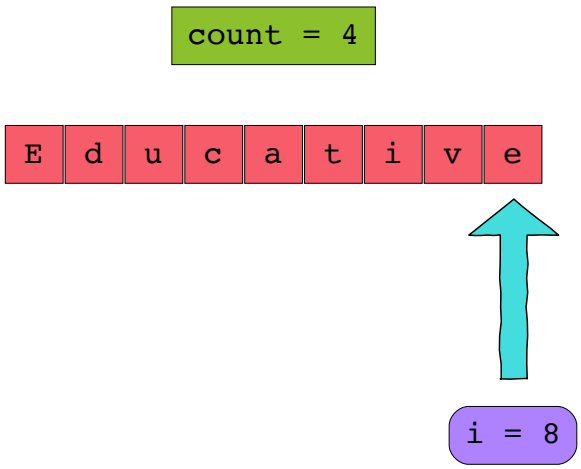
Iteration 5



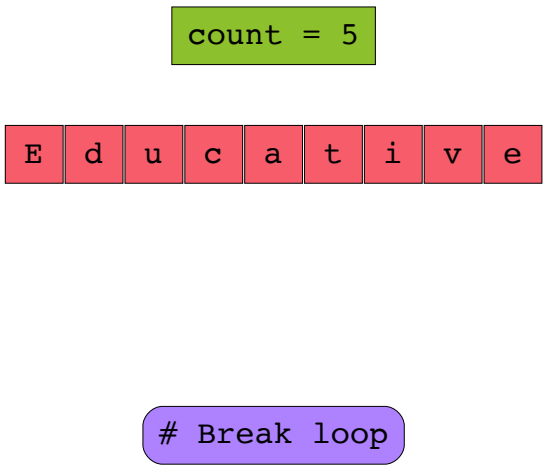
Iteration 6



Iteration 7



Iteration 8



Breaking the loop and returning the variable count

Let's take a look at the code:

```
def isVowel(character): # function to check whether input character is a vowel
  character = character.lower() # convert character to lower case so upper cases can also
  vowels = "aeiou" # string containing all vowels

  if character in vowels: # check if given character is in vowels
    return 1
  else:
      return 0

def countVowels(string): # function that returns the count of vowels
  count = 0
  for i in range(len(string)):
    if isVowel(string[i]) : # check if character is vowel
      count += 1
  return count

# Driver code
string = "Educative"
print(countVowels(string))
```

Iterative method for calculating number of vowels in a string

In the code snippet above, the main function `countVowels()` takes a `string` as its input. It traverses the entire length of the string. The condition

```
i in range(len(string))
```

ensures that the `for` loop breaks if `i` becomes equal to the length of the string calculated using `len(string)`. Next, we use a condition

```
isVowel(string[i])
```

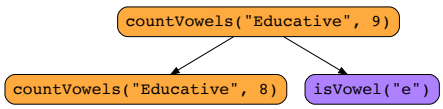
to determine whether or not that particular character is a vowel. If this condition is satisfied, we increment `count` by 1.

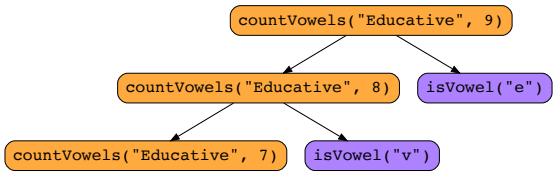
`isVowel()` is a function that returns `TRUE` if the character it passes is a vowel. It returns `FALSE` if that character is not a vowel.

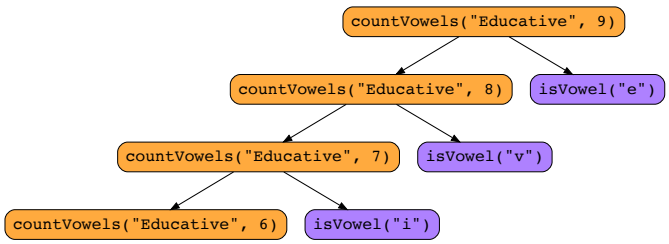
Recursive Method

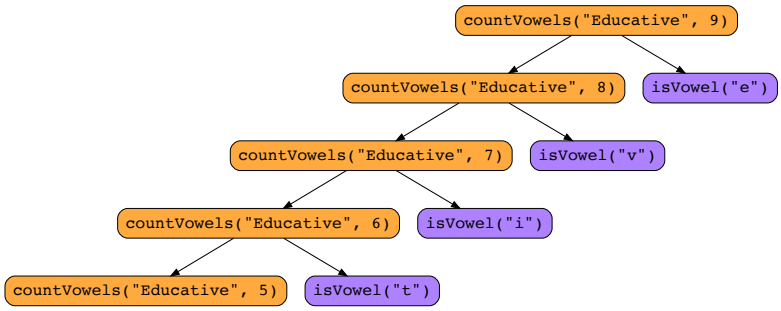
Let's take a look at the recursive counterpart:

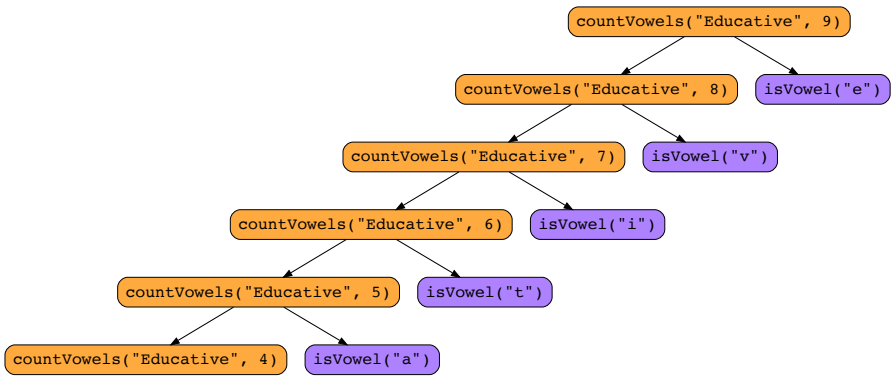
countVowels("Educative", 9)

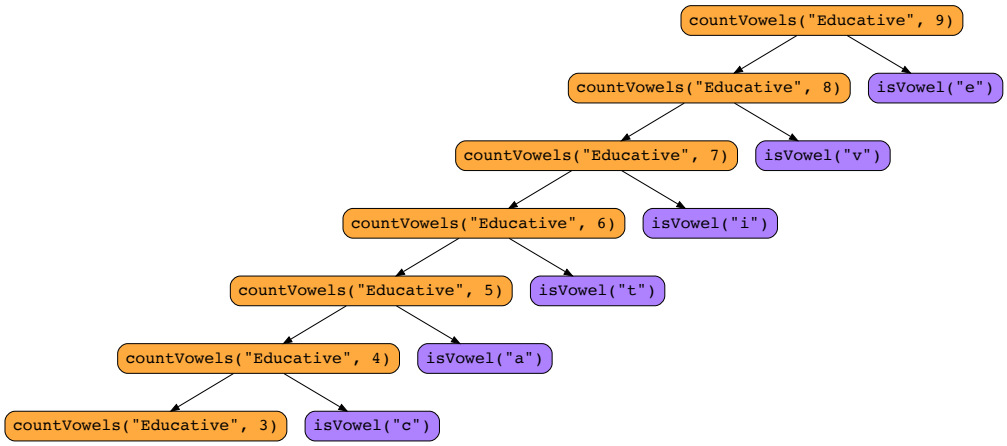


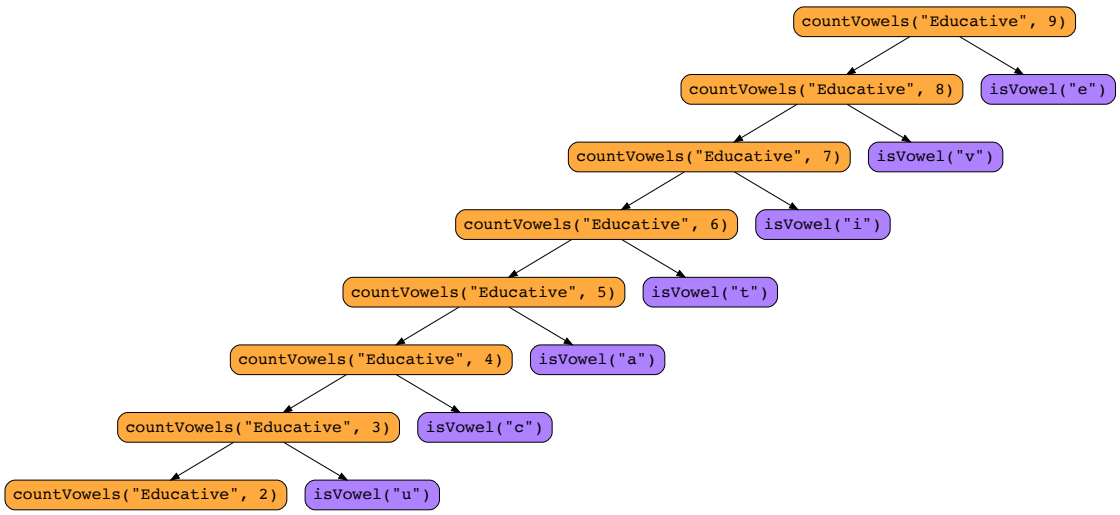

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")

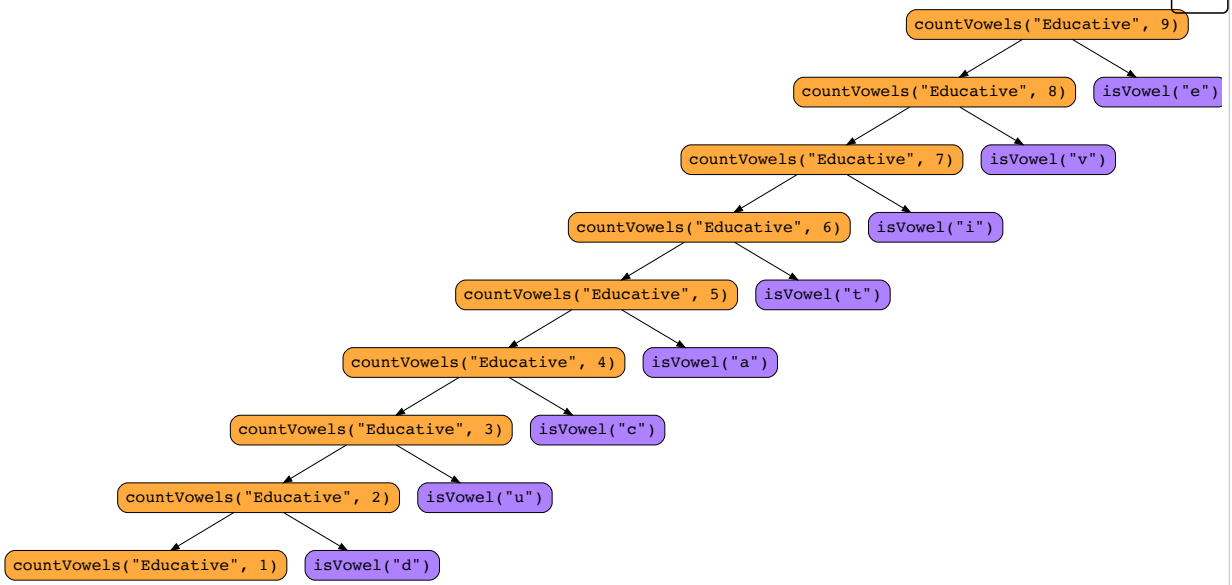

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")

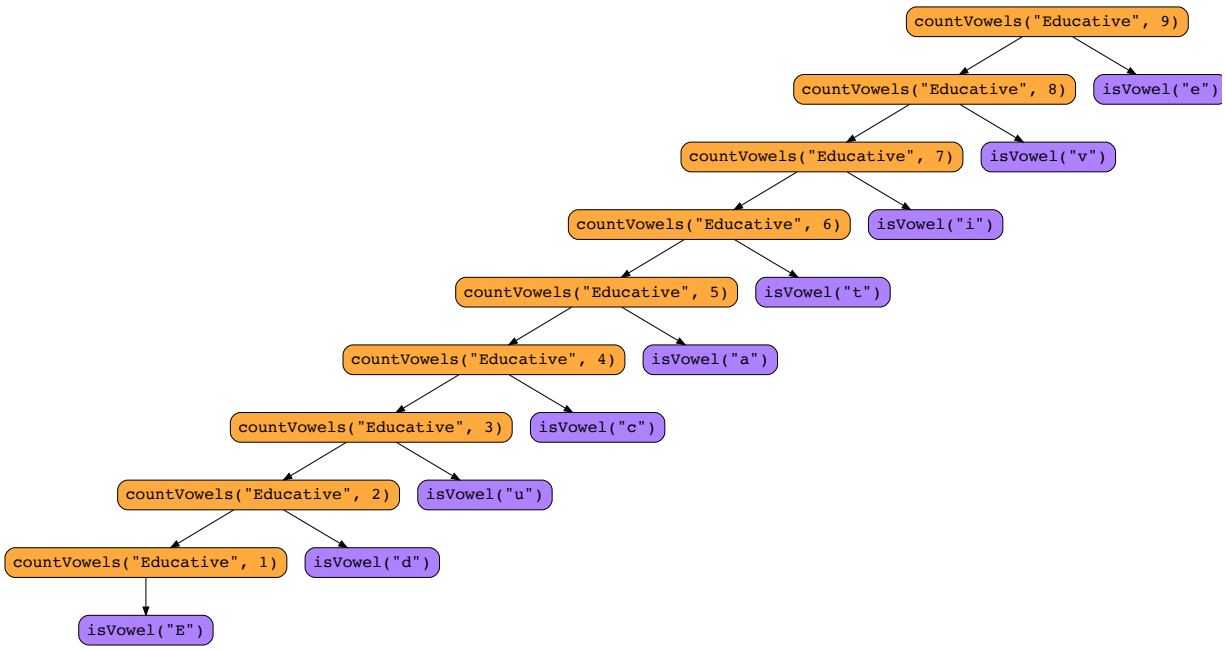

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")

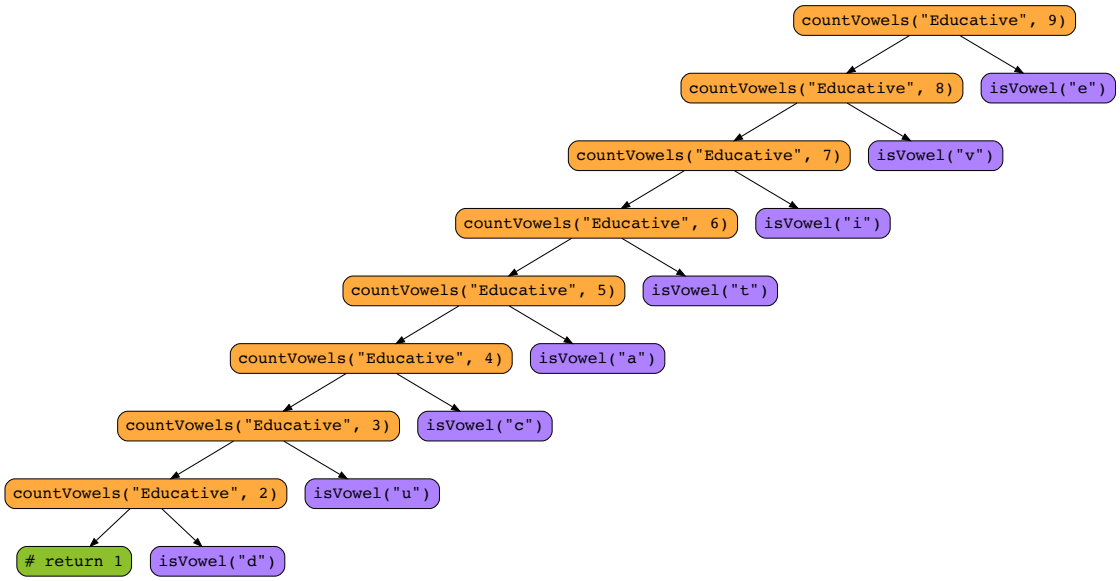

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
countVowels("Educative", 4)
isVowel("a")

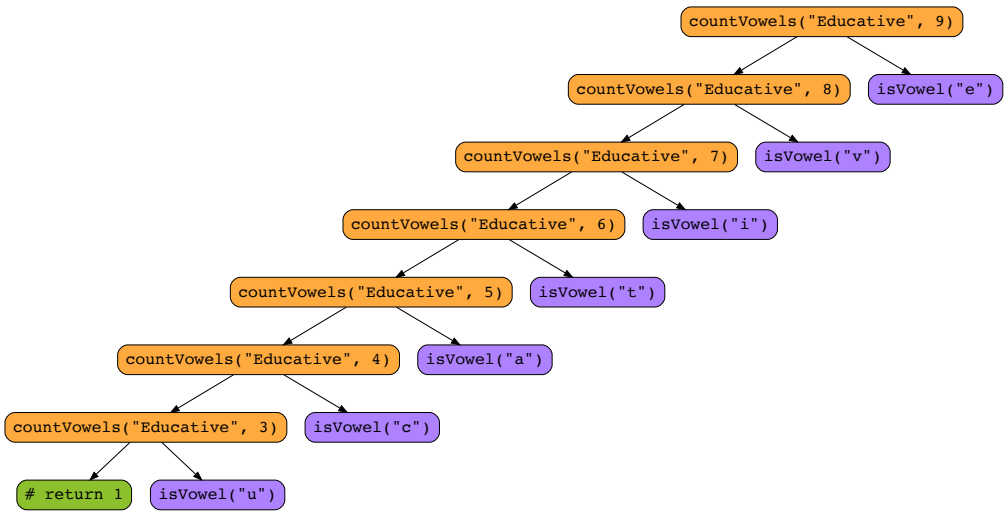

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
countVowels("Educative", 4)
isVowel("a")
countVowels("Educative", 3)
isVowel("c")

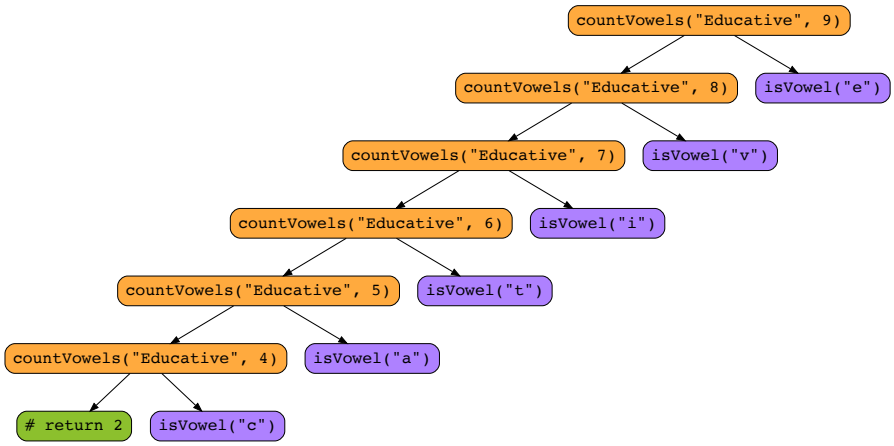

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
countVowels("Educative", 4)
isVowel("a")
countVowels("Educative", 3)
isVowel("c")
countVowels("Educative", 2)
isVowel("u")

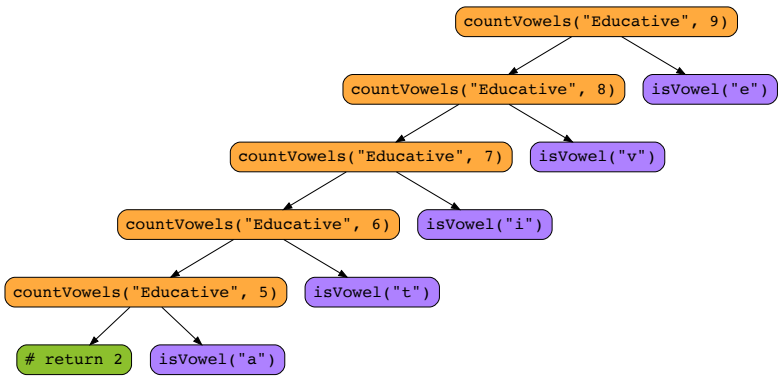

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
countVowels("Educative", 4)
isVowel("a")
countVowels("Educative", 3)
isVowel("c")
countVowels("Educative", 2)
isVowel("u")
countVowels("Educative", 1)
isVowel("d")

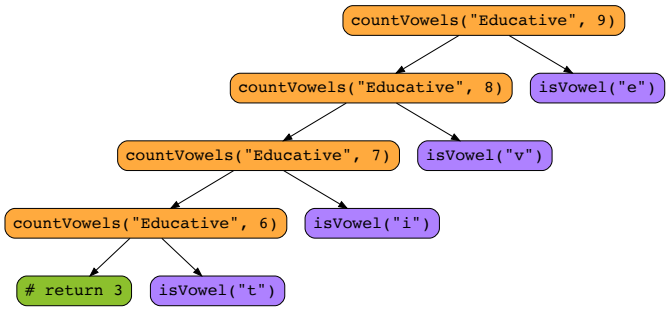

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
countVowels("Educative", 4)
isVowel("a")
countVowels("Educative", 3)
isVowel("c")
countVowels("Educative", 2)
isVowel("u")
countVowels("Educative", 1)
isVowel("d")
isVowel("E")

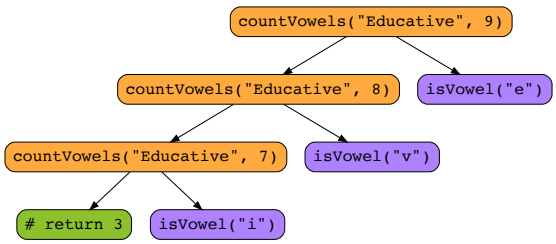

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
countVowels("Educative", 4)
isVowel("a")
countVowels("Educative", 3)
isVowel("c")
countVowels("Educative", 2)
isVowel("u")
# return 1
isVowel("d")

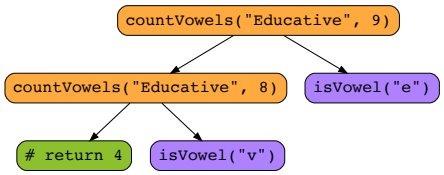

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
countVowels("Educative", 4)
isVowel("a")
countVowels("Educative", 3)
isVowel("c")
# return 1
isVowel("u")

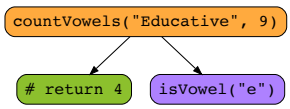

countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
countVowels("Educative", 4)
isVowel("a")
# return 2
isVowel("c")


countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
countVowels("Educative", 5)
isVowel("t")
# return 2
isVowel("a")


countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
countVowels("Educative", 6)
isVowel("i")
# return 3
isVowel("t")


countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
countVowels("Educative", 7)
isVowel("v")
# return 3
isVowel("i")


countVowels("Educative", 9)
countVowels("Educative", 8)
isVowel("e")
# return 4
isVowel("v")


countVowels("Educative", 9)
# return 4
isVowel("e")

```
# return 5
```

19 of 19

The code will look something like this:

```
def isVowel(character): # function to check whether input character is a vowel
  character = character.lower() # convert character to lower case so upper cases can also

  vowels = "aeiou" # string containing all vowels

  if character in vowels : # check if given character is in vowels
    return 1
  else:
      return 0

def countVowels(string, n): # function that returns the count of vowels
  # Base Case
  if n == 1 :
    return isVowel(string[0])

  # Recursive Case
  return countVowels(string, n - 1) + isVowel(string[n - 1])

# Driver code
string = "Educative"
print(countVowels(string, len(string)))
```

Recursive method for calculating number of vowels in a string

The `isVowel()` function checks whether or not the character being input to it is a vowel. The major change that occurs is the conversion of the `for` loop into the recursive call.

Each time, the length of the input `string` is reduced in the iterative method. This is also done in the recursive method. We reduce the length of the input `string` and pass another call of the same function. The leftover letter is examined to check if it is a vowel. The result is added in the function calls.

In the next lesson, we have a challenge for you to solve.

Back

Converting Iterative Code to Recursive...

Next

Challenge 1: Compute Square of a Nu...

Mark as Completed

Report an Issue

Ask a Question (https://discuss.educative.io/tag/count-vowels-in-a-string__iteration-vs-recursion__recursion-for-coding-interviews-in-python)