COMP10001 Foundations of Computing
Semester 1, 2019

Tutorial Questions: Week 8

— VERSION: 1474, DATE: APRIL 10, 2019 —

Discussion

1. What is a "bug"? What are some debugging strategies?
2. What are the three types of errors we've learned about and what do they mean?

   Now try Exercises 1 & 2
3. What is "lazy evaluation"? How can it make our code safer and more efficient?
4. What is "`None`"? How is it used?

   Now try Exercise 3

Exercises

1. Imagine we are testing a function which has been written. It's supposed to take an employee's ID and a year as input, fetch their records and calculate the total amount of tax which they paid in that year. Describe five aspects of the code's functionality you would test if you were to write test cases.
2. Find the errors in the following programs. Classify them as (a) syntax, (b) runtime or (c) logic errors.

   (a)
```
def disemvowel(text):
    """ Returns string `text` with all vowels removed """
    vowels = ('a', 'e', 'i', 'o', 'u')
    answer = text[0]
    for char in text:
        if char is not in vowels:
            answer = char + answer
    print(answer)
```

   (b)
```
def big-ratio(nums, n):
    """ Calculates and returns the ratio of numbers
    in list `nums` which are larger than `n` """
    n = 0
    greater_n = 0
    for number in nums:
        if number > n:
            greater_n += 1
            total += 1
    return greater_n / total

nums = [4, 5, 6]
    low = 4
print(f"{100*big_ratio(nums, low)}% of numbers are greater than {low}")
```

3. Compare the two functions below. Are they equivalent? Why would we prefer one over the other?

```
def noletter_1(words, letter='z'):
    for word in words:
        if letter in word:
            return False
    return True

def noletter_2(words, letter='z'):
    no_z = True
    for word in words:
        if letter in word:
            no_z = False
    return no_z

wordlist = ['zizzer'] + ['aadvark'] * 1000000
print(noletter_1(wordlist))
print(noletter_2(wordlist))
```

Problems

1. Write a program which asks the user to enter a series of words and then checks whether any of those words are palindromes (spelled the same way backwards as forwards, like hannah). You should print True if there are any palindromes and False if there are none. Use lazy evaluation to save some time!

2. Write a function which takes a string, a character and an integer threshold and returns `True` if the character appears in the string with a frequency above the threshold, `False` if it appears below the threshold and `None` if it doesn't appear at all.