

1. Write a recursive function that reverses a string (e.g. “Don’t get sick” yields “kcis teg t’noD”).

2. Perform a substitution trace on
`reverse('Cinco-fone')`

3. What does the following evaluate to?

```
1 def writeThatDown( n ):
2     if n < 5:
3         return n
4     return (2 * n)
5
6 def he( n ):
7     temp = n + 180
8     if temp > 185:
9         return temp
10    return n
11
12 def putstheFernback( n ):
13     return -n
14
15 n = 20
16 n = he(putstheFernback(writeThatDown( n ) ) )
17 print( n )
```

- You should be able to shift a string by a value greater than the length of the string¹. Assume a function `len(str)` which returns the length of a string is provided.

- (d) Implement the function `shift_right()`, which rotates letters in the opposite direction.

2

5. Write a function that takes in a file name, and returns the average size of a word in that file. The files will only have 1 word per line, for example:

No
soup
for
you!

which has an average length of: 3.25

Assume a function `len(str)` which returns the length of a string is provided.

6. Write a function that takes in a string representation of a number and returns the sum of all of digits in the string.

For example `sum('11111')` returns 5.

Assume a function `len(str)` that returns the length of a string is provided.

Further assume there's a function `int(str)` which, given a string representation of an integer, returns its integer value.

(a) Recursively.

(b) Iteratively

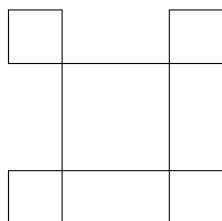
(c) How would you test this function?

7. Assuming the turtle is facing East, write the Python code to draw the following picture given the proper depth as input:

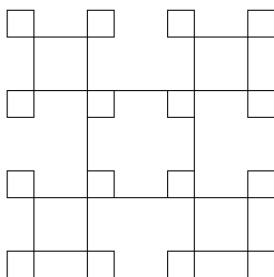
- depth = 0
No output
- depth = 1



- depth = 2



- depth = 3



8. Raymond and Connor have an issue. Raymond is a sadist and loves destroying or changing strings. Connor, on the other hand, loves keeping a count of things. You will be writing a function to satisfy both of their desires.
- (a) Write a `strhead()` function, which takes a string and returns the character in the first index of the string.
 - (b) Write a `strtail()` function, which takes a string and returns everything in the string except for the first index of the string (everything after the head).
 - (c) Write a tail recursive function `coRec`, which takes a string and a character and returns the number of times the character appears in the string. For example, `coRec("Eric is enjoying the weather.", 'i')` should return 3. Do not use the `len()` function.
 - (d) Write an iterative function `rmIter`, which takes the same parameters above and returns the string with all instances of the character removed. For example, `rmIter("Chris has hats.", 'a')` should return "Chris hs hts."

- (e) Write an iterative function (using a for/while loop, etc.) `rmNco`, which takes a string and a character, removes all instances of the character in the string, prints the resulting string, and lastly returns the number of times a character was removed. You may use the `len()` function. For example, `rmNco("We collectively enjoy things", 'o')` prints "We collectively enjy things", and returns 2.
- (f) What are the complexity of the functions you wrote for parts 8c, 8d, and 8e?
9. Write a function which performs a basic string compression, which uses counts of repeated characters. For example the string `abbbccccaaa` would be compressed to: `a1b3c4a3`.
Assume a function `len(str)` which returns the length of a string is provided.
assume a function `int(str)` which, given a string representation of an integer, returns its integer value.
Assume a function `str(int)` which, given an integer, returns the string representation of this integer, is provided.

10. Sally is being plagued by an army of lookalike suiters, each of which presents her with an enticing but unbearable meal upon arrival. The dishes all smell amazing, however, so she can't help but try each one. A dish will never fail to disappoint her, but fortunately, some of the suiters shared recipes and created identical concoctions. Once Sally tastes a meal once, she can immediately smell out instances of the same dish and send their bearers away.

(a) Write a Python function that, given a list of “dishes,” returns a list of the rejected dishes in the order they failed to fool her. (e.g. `[1, 2, 2, 3, 3, 3]` would return `[2,3,3]`)

(b) If your solution to 10a was iterative, write it recursively; if it was recursive, write it iteratively.

(c) What is the time complexity of your approach? Why?

11. For the sake of this question, you find yourself to be the head programmer under Kim Jong Un's glorious rein. It also just so happens that a nation-wide track meet is being held today. Thus, the glorious leader has demanded that you write a program to keep track of information relating to all track runners present at the event.
- (a) Write a class named `TrackRunner` to keep track of all runners competing. You will need to store their `name`, `age`, and `fastestTime`.
 - (b) Now write a function to make an individual `TrackRunner` object.
 - (c) The glorious leader has decided that, on this day, no runner named `Joe` may win gold. Given a list of `TrackRunner` objects, write the function `aWinnerIsYou(r)` that finds the runner in the list `r` with the fastest time who's name is not `Joe`. Then print the runner's name, age and time.

12. Below is Python code for a function that performs an insertion sort and prints `data` after each iteration of the `for` loop.

```
1 def insertion_sort( data ):
2     for marker in range( 1, len( data ) ):
3         val = data[marker]
4         i = marker
5         while i > 0 and data[i-1] > val:
6             data[i] = data[i-1]
7             i -= 1
8         data[i] = val
9         print( data )
```

(a) Write out what the function will print for the input list: `[3,2,7,1]`.

(b) What is the sort algorithm's time complexity?

13. **Greedy algorithms.** In the game Black and White², the player is faced with a row of identical double-sided chips. You can probably guess what colors the two sides of each chip are. The objective is to flip as many chips as necessary so their exposed colors match that of a target pattern.

The catch? Reordering the chips is said to be Impossible by those who seem to know what they're talking about.

The *real* catch? Flipping a group of consecutive tiles can be accomplished in a single "action."

If every flip takes one "action," write a function `bwMoves` that, given a starting pattern and target pattern as equal-length strings, returns the minimum number of actions required to get them to match. For instance, `bwMoves('BBWBBWBBBB', 'WWWWBBWBB')` should return 3.

Hint: Were any of the other questions labeled with the concept they were testing?

²Special thanks to Professor Butler for unwittingly allowing us to rip off his problem.