

CS 515 Homework 4

Jared Bass

February 2022

Homework 4: Recursion
Due: 2/23/22 at 11:59pm

1 Introduction

For this homework, write the following Python functions using Recursion. You may use recursion, conditional statements (`if`, `else`, `elif`), list or string indexing and slicing. Some of these problems can be written without using recursion, e.g. using `map`, `filter`, `reduce`, or other looping structures. However, the objective here is to build your recursion muscles, so please stick to **Recursion**!

Do not use built-in functions (e.g. `len`, `sum`, etc). However, your functions may call other functions that you write yourself. Be sure to include a doc-string/specification under the signature line of each function. The specification should indicate what the function computes (return value) and what are the preconditions of the inputs.

Please be sure to name your functions exactly as specified for TA grading.

Please submit this homework as a single file containing all functions, called `hw4.py`

2 Question 1: Implement the Dot Product (20 points)

Write the function `dot(L,K)`. The input of this function is two lists, `L` and `K`. Recall that the dot product of two vectors or lists is the sum of the products of the elements in the same position in the two vectors.

The dot product is only defined for two lists of equal length. If the lengths of the lists are not equal, please return negative infinity. (You can look up how this is done in python. HINT: It uses the `float` function)

If these two lists are both empty, dot should output 0.0. Assume the input lists contain only numeric values.

```
>>> dot([5,3], [6,4]) <- Note that 5*6 + 3*4 = 42
42
```

3 Question 2: Write a String Explosion function (20 points)

Write a function called `explode(S)` that takes a string as input and returns a list of all of the characters in the string. For example:

Here is the `explode` function in action:

```
>>> explode('spam')
['s','p','a','m']
>>> mean("")
[]
```

4 Question 3: Write your own index function (25 points)

Write a function `index(e,L)` that takes in an element `e` and a sequence `L` where by "sequence" we mean either a list or a string. Note that indexing and slicing works the same for both lists and strings, so your function should be able to handle **both** types of input!

Then, `index` should return the index at which `e` is **first** found in `L`. Counting begins at 0, as is usual with lists. If `e` is NOT an element of `L`, then the function should return an integer that is equal to the length of `L`. **Don't use** the built-in `len` function explicitly! Your recursive implementation can do this by itself

NOTE: You will receive NO POINTS if your function just calls the pre-existing implementation of `index` or `find`. You MUST write this code yourself.

```
>>> index(42, [ 55, 77, 42, 12, 42, 100 ])
2
>>> index(42, range(0,100))
42
```

```

>>> index('hi', [ 'hello', 42, True ])
3
>>> index('hi', [ 'well', 'hi', 'there' ])
1
>>> index(' ', 'outer exploration')
5

```

NOTE: As you can see, this function is NOT called like list index. It is explicitly called by itself from the command line. DO NOT write a function that does `L.index(e)`. This will receive NO POINTS!

5 Question 4: Write a deep reversal function (35 points)

Write a function called `deepReverse(L)` that takes as input a **list** of elements `L` where some of the elements of the list may be lists themselves. `deepReverse` returns the reversal of the list where additionally, any elements within `L` that are also lists are `deepReversed` as well. Here are some examples:

```

>>> deepReverse([1, 2, 3])
[3, 2, 1]
>>> deepReverse([1, [2, 3], 4])
[4, [3, 2], 1]
>>> deepReverse([1, [2, [3, 4], [5, [6, 7], 8]]])
[[[8, [7, 6], 5], [4, 3], 2], 1]

```

For this problem, you will need the ability to test whether or not an element in the list is a list itself. You might use the following line of code to test whether or not `x` is a list:

```

if isinstance(x, list):
    # if True you will end up here
else:
    # if False you will end up here

```

Again, please submit this homework as a single file containing all functions, called `hw4.py`