

# CSC148: Recursion

Daniel Zingaro  
University of Toronto

January 2016

## Question 1

Here is a slight change to the recursive solution of the binary codes example that we studied in lecture. Is this new version still correct?

```
def codes(r):  
    '''(int) -> list of str  
  
    Return all binary codes of length r.  
    '''  
    if r == 0:  
        return ['']  
    small = codes(r-1)  
    lst = []  
    for item in small:  
        lst.append('0' + item)  
        lst.append('1' + item)  
    return lst
```

## Question 2

Consider the following code.

```
def mystery(lst, k, value):  
    if len(lst) == 0:  
        return k <= 0  
    if lst[0] == value:  
        return mystery(lst[1:], k-1, value)  
    else:  
        return mystery(lst[1:], k, value)
```

Complete the following sentence:

If `mystery` is run, then the goal that it accomplishes is to ... (i.e. we want a brief, English description of what the function does, not a line-by-line explanation of the code).

## Question 3

The following code tries to accomplish something potentially useful ... although it has a bug! What is the goal of the function and why is it incorrect?

```
def skip_sum(n):
    if n == 0:
        return 0
    return n + skip_sum(n-2)
```

## Question 4

Complete the following function using recursion.

```
def contains(lst, value):
    '''(list of int, int) -> bool

    Return whether or not lst contains the given value.
    lst may have nesting to arbitrary depth.

    >>> contains([1, [[2], 3]], 3)
    True
    >>> contains([1, [[2], 3]], 4)
    False
    '''
```

## Question 5: Challenge

Here is a bigger change to the recursive solution of the binary codes example that we studied in lecture. Is this new version still correct?

```
def codes(r, current, code_list):
    '''(int, str, list of str) -> list of str

    Return all binary codes of length r.
    '''
    if r == 0:
        code_list.append(current)
        return code_list
    codes(r-1, current + '0', code_list)
    codes(r-1, current + '1', code_list)
    return code_list
```