

# 15-110 Recitation Week 5

## Reminders

- [Recitation Feedback Form](#)
- Check 3 due Monday 2/19 @ Noon
- Check 2 and HW 2 revisions due 2/20 @ Noon
- Exam on 2/21
  - Review Sessions
  - Small Groups!
  - Also OH and Piazza are always there if you need individual help!

## Overview

- List methods
- Recursion
- Aliasing
- Recursion (code writing)

# Problems

## LIST CODE WRITING: REMOVE MATCHES

Write a function `removeMatches(L, matchList)` that takes in a list of numbers `L`, and removes all of the elements in `L` that are also in `matchList`. Write this function both destructively and non-destructively

For example, `removeMatches([1,2,3,4,5],[1,5,10,15])` should return `[2,3,4]` and `L = [1,2,3,4,5]`

And `destructiveRemoveMatches(L,[1,5,10,15])` returns `none`, but `L = [2,3,4]`

Destructive:

Non-Destructive:

--	--

## RECURSION INTRO

General notes on recursion:

--

Recreate the following function using recursion (write on the right empty space):

```
def double(lst):  
    result = []  
    for i in range(len(lst)):  
        result.append(2 * lst[i])  
    return result  
  
#double([1,2,3]) -> [2,4,6]
```

--

## LIST ALIASING

Code trace and compare the following two options for ways to create “empty” 2D lists:

Option 1:

```
inner = [0, 0, 0, 0]
outer = [inner, inner, inner]
```

Option 2:

```
rows = 3
outer = []
for row in range(rows):
    outer.append([0, 0, 0, 0])
```

**For each option**, after running the code above, what are the values in outer?

Option 1: outer =

Option 2: outer =

After adding the following line of code and running it:

```
outer[0][0] = 42
```

What are the values in outer?

Option 1: outer =

Option 2: outer =

Be sure you can explain what difference you are seeing, and which option you should use and why.

## RECURSIVE CODE WRITING

Write the function `sumOddMToN(m, n)` that takes two integers and recursively calculates the sum of all odd integers between `m` and `n`, **not including `n`**. You are guaranteed that  $m > 0$  and  $n > 0$ , and  $m < n$ .

Example: `sumOddMToN(3,10)` should return 24, as  $3+5+7+9=24$ , while `sumOddMToN(2,7)` should return 8 as  $3+5=8$  (7 is not inclusive).