15-112 Midterm 2 Review

Up to 75 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use try/except on this review.

1. (points) CT1

```
def ct1(L):
    s = set()
    for v in L:
        if isinstance(v, dict):
            for k in v:
                s.add(v[k])
        else:
            for e in v:
                if e in s:
                    s.remove(e)
        print(s)
    return s

print(ct1(['CB', {2:'A', 'B':3, 4:'C'},
           'BC', [4, 3, 2]]))
```

2. (points) CT2

```
import copy

def ct2(L, A, n):
    A[-1][0] += n
    A[n%2] += [10*n]
    L.append(n)
    print(f'A: {A}')

L = [[3], [4]]
C = copy.copy(L)
D = copy.deepcopy(L)
ct2(L, C, 1)
ct2(L, D, 2)
print(f'L: {L}')
```

3. (points) **Free Response: Recursive getHiLo**

   Write the recursive function `getHiLo(lst)` which, given a list of integers `lst` returns a tuple contain the highest and lowest values in the list. For example:

   `getHiLo([1, 7, 3, 8, 2, 9, 6, 4, 5])` returns `(9,1)`

   `getHiLo([5])` returns `(5,5)`

   `getHiLo([])` returns `None`

   Your solution must use recursion. If you use any loops, comprehensions, or iterative functions, you will receive no points on this problem.

4. (points) **Free Response: Find Zero Triplets**

Write the function `findZeroTriplets(L)` that takes as input a list `L` of integers of length `N` and returns a set of all triplets in the list whose sum is equal to 0. For example, if the given list is `[-1, 0, -3, 2, 1]`, you should return `{(-1, 0, 1), (-3, 1, 2)}` (note that triplets are sorted). If there is no valid triplet, you should return the empty set. You may assume that `L` is a list containing only integers. The *naive* solution would use 3 loops to check all triplets of values in `L`.

You should use sets and/or dictionaries to do this faster than $O(N^3)$

You should also specify the Big-O of your solution in the box below:

5. (points) **Big-O:** For the functions shown below, write the total Big-O runtime of the function in terms of N, the length of the function argument, in the box to the right of the code. All answers must be simplified- do not include lower-order terms!

```
def f1(L): # L contains N items   #BigO
    M = sorted(L)                 #------
    count = 0                     #------
    for elem in M:                #------
        count += M.count(elem)    #------
        #update elem              #------
    return count                  #------
```

```
def f2(L): # L contains N items   #BigO
    newL = []                     #------
    n = len(L)                    #------
    for i in range(0, n, n//4):   #------
        newL.append(L.pop())      #------
        #update i                 #------
    return newL                   #------
```

```

def f3(s): # s contains N characters   #BigO
    if s.isdigit():                    #------
        letters = set(s)               #------
        for c in letters:              #------
            if c == '4':               #------
                return "YES"           #------
            #update c                  #------
    return "NO"                        #------
```