CSC236 Week 10 Tutorial:

# Review for Term Test 2

Slides Credit: Charlene Chu

# Exercise 1

This question concerns the following code.

```
def mystery(lst):
    '''Pre:  lst is a list of 0s and 1s.
    Post:  ...
    '''
    i = 0
    j = 0
    while i < len(lst):
        if lst[i] == '0':
            j = j + 1
        else:
            j = j - 1
        i = i + 1
    return j == 0
```

a. Is `j` a variant for this loop? Explain why or why not.

b. State and prove a loop invariant for this loop. Your invariant should be useful for proving the postcondition. Prove only the loop invariant, not loop termination or the postcondition.

c. State and prove the postcondition for `mystery`. Your postcondition should characterize what the algorithm does.

# Exercise 2

```
def count_ordered ( A ):
    '''
    Pre:  A is a list of numbers
    Post:  Outputs the number of pairs (i,j)
    such that i<j and A[i]<=A[j]
    E.g., count_ordered([3,2,5,1])=2
    and count_ordered([10,10])=1
    '''
    if len(A)==0:
        return 0
    else:
        count=0
        i=1
        while i<len(A):
            if A[i]>=A[0]:
                count+=1
            i+=1
    return count+count_ordered(A[1..len(A)-1])
```

Read the following program's specifications and code carefully. Notice both the loop and the recursive call.

a. State and prove a helpful loop invariant for the `while` loop ("steps 1 and 2" from lecture). You may assume $\texttt{i} \in \mathbb{N}$ and $\texttt{i} \leq \texttt{len(A)}$ have already been proven as loop invariant (so use them freely).

b. Using your loop invariant from part (a), prove the correctness of `count_ordered` according to the specifications. Once again, note that `count_ordered` is recursive, so you need to analyse a recursive call.