CS210 HW3 Answer Sheet
#1
best: Array is in descending. Flag is never set.
worst: Array is ascending. Flag is set every time.

Array Access:
best: 2(n-1) // j goes from 0 to n-2 and each time there are only two array accesses if the the compare is always false

worst: 2(n-1)n + 4(((n-1)n)/2) = 4(n-1)n
2(n-1)n is 2(n-1) from best case and then multiplied by n because we set flag=true n times

4(((n-1)n)/2) is 4 times the number of times the if statement is true. Which is (n choose 2) aka n(n-1)/2
We can look at this for n=5 with an ascending input. Every time we move one element we make n-1 exchanges (each row). Flag is set n times so we have 5 rows. The final row never returns true.

input = [1, 2, 3, 4, 5]
aa=6 aa=12 aa=18 aa=24, flag=true, [2, 3, 4, 5, 1]
aa=30 aa=36 aa=42 aa=44, flag=true, [3, 4, 5, 2, 1]
aa=50 aa=56 aa=58 aa=60, flag=true, [4, 5, 3, 2, 1]
aa=66 aa=68 aa=70 aa=72, flag=true, [5, 4, 3, 2, 1]
aa=74 aa=76 aa=78 aa=80

Compares:
best: n + n-1 // we do n compares on the for loop and n-1 on the if statement
worst: n^2 + n(n-1)

n^2 // we run the for loop n times and each time costs n compares
n(n-1) // we execute this as the inside of the for loop. We save n compares because we are inside the loop and don't have to test to break out of it.

Costs for various n:
n=0, aa=0 c=0
n=1, aa=0 c=1
n=2, aa=8 c=6
n=3, aa=24 c=15
n=4, aa=48 c=28
n=5, aa=80 c=45
n=6, aa=120 c=66
n=7, aa=168 c=91
n=8, aa=224 c=120

#2
best: Number has a non-9 in the ones place
worst: Number is all 9's

Array:
best: 2
worst: 2n +1 +2n

Compares:
best: 1 // or 2 if counting ==
worst: n+1 + n+1 // or 2n+1 + n+1 if counting ==