# CSC236 winter 2020, quiz week $6_1$

**first/given name:** **last/family name:**

**utorid:**

Use induction to prove the correctness of the function below, with respect to the precondition and postcondition stated in the docstring. (You do not need to restate or further formalize the given pre- and postcondition - though you may do so if you find it helpful.)

(Note that '%' is the modulo operator in Python. So for any integer k, `k % 2` is the parity of k, i.e. the remainder mod 2.)

```
def total_parity(A):
  """Pre: A is a list only containing integers
  Post: Return the parity of the sum of values in A. i.e. return 1 if the
  sum is odd, and 0 if it is even.
  """
  if len(A) == 0:
    return 0
  tail_parity = total_parity(A[1:])
  return (A[0] + tail_parity) % 2
```

**Solution** Basis: Consider a list $A$ of length 0. The sum of $A$ is 0, meaning our function should return 0, which it does, by the first 2 lines.

Inductive step: Let $n \in \mathbb{N}$ and assume the function is correct on inputs of size $n$. I will show that it is therefore correct on inputs of size $n+1$.

Let $A$ be a list of length $n+1$ satisfying the precondition.

Let $s$ be the sum of the last $n$ elements of $A$. By our IH, `total_parity(A)` will return the parity of $s$.

In other words, $\exists k \in \mathbb{N}, s = 2k + \texttt{tail\_parity}$

Let $s' = s + A[0]$, i.e. the sum of elements in $A$. Then $s' = 2k + \texttt{tail\_parity} + A[0]$

Clearly we can subtract $2k$ from $s'$ and the result will have the same parity. So the parity of $s'$ is equal to the parity of $\texttt{tail\_parity} + A[0]$, which is exactly what our function returns.

Thus, our function is correct on inputs of size $n+1$, completing the induction.