

## Assignment

### 1. Sum of Numbers:

*Algorithm:*

1. Initialize a variable `sum` to 0.
2. Loop through each element `num` in the list or array.
3. Add the current `num` to `sum`.
4. After the loop ends, the variable `sum` will contain the total sum of the elements.
5. Return the value of `sum`.

```
function sum(n):  
    if n == 0:  
        return 0  
    else:  
        return n + sum(n - 1)
```

### 2. Largest Element:

*Algorithm:*

1. If the list is empty, return `None` (or another indicator for an empty list).
2. Initialize a variable `max_num` with the value of the first element of the list.
3. Loop through each element `num` in the list starting from the second element:
  - o If `num` is greater than `max_num`, update `max_num` to `num`.
4. After the loop ends, `max_num` will hold the largest value in the list.
5. Return `max_num`.

```
function find_largest(arr, size):  
    if size == 1:  
        return arr[0]  
    else:  
        max_subarray = find_largest(arr, size - 1)  
        return arr[size - 1] if arr[size - 1] > max_subarray  
    else  
        max_subarray
```

### 3. Even/Odd Numbers Count:

*Algorithm:*

1. Initialize two counters, `even_count` and `odd_count`, to 0.
2. Loop through each element `num` in the list.
3. If `num % 2 == 0`, increment `even_count`.
4. Otherwise, increment `odd_count`.
5. After the loop ends, both `even_count` and `odd_count` will hold the counts of even and odd numbers, respectively.
6. Return both `even_count` and `odd_count`.

```
function print_even_odd(start, end):
```

```
    if start > end:
```

```
        return if start % 2 == 0:
```

```
    print(start)
```

```
    print_even_odd(start + 1, end)
```

## 4. Digit Counting:

*Algorithm:*

1. Initialize an array `digit_count` of size 10 with all values set to 0 (to count digits 0 through 9).
2. Loop through each number in the list:
  - Convert the number to a string to access each digit.
  - For each digit in the string, convert it back to an integer and increment the corresponding index in the `digit_count` array.
3. After the loop ends, `digit_count` will contain the counts of each digit (from 0 to 9).
4. Return `digit_count`.

```
function count_digits(n):  
    if n == 0:  
        return 0  
    else:  
        return 1 + count_digits(n // 10)
```