NAME _____ **ICA-11**

NetID_____ @email.arizona.edu

Work with your neighbor.

---

1. Write a recursive function `sumlist(L)` that returns the sum of the elements in L.

   What is the base case?

   What is the recursive case?

2. Write a recursive function `string_len(s)` that returns the length of string `s`.

   What is the base case?

   What is the recursive case?

3. Write a recursive function that `join_all(alist)` that takes a list `alist` and returns a string consisting of every element of `alist` concatenated together.

   What is the base case?

   What is the recursive case?

4. Write a recursive function that implements `join`. That is, write a function `join(alist, sep)` that takes a list `alist` and returns a string consisting of every element of `alist` separated by the string `sep`.

   What is the base case?

   What is the recursive case?

5. Write a function `even_positions(s)` that returns a string consisting of all the even-numbered positions of the Python string s.

What is the base case?

What is the recursive case?

6. Write sumlist() in the two ways described in the exercise.

- Options:
    - Recurse on L[:-1] and then add in L[-1]
    - Recurse on each half and add the results

```
class Node:
    def __init__(self, value):
        self._value = value
        self._next = None
```

```
class LinkedList:
    def __init__(self):
        self._head = None

      def add(self, new):
        new._next = self._head
        self._head = new
```

7. Consider a linked list whose value attributes consist of integers. Using the `Node` and `LinkedList` class definitions above, write a method `count_odds(self)` that returns a count of the *number of elements* whose `_value` attributes are odd. If the list is empty, the method returns `0`.

8. Describe the stack ADT. What are its main properties? What are the operations that can be performed on a stack?