1. (8 points)
   a. \([1, -6, 'h', 3][2] \times (10/3)\)
   b. \([1] + [[3], 'a', [4, 5]]\)
   c. \(([[1, 2], [3, 4], [5, 6]][2] + \{-1: [1], 0: [2], 1: [3]\}[0])\)
   d. \(((\text{len}([1, 2, 3, 4]) + \{1, 2, 3\}[1]) > 6) \text{ and } (\text{not ('E' in 'exam'))}\)

2. (8 points) What value does `foo()` return?

   ```python
def bar(a, b, c):
    a = 7 - b
    c[0] = a + 15
    c = [8]
    return [a, b, c]

def foo():
    a = 4
    b = a
    c = [1, 2, b]
    temp = bar(a, b, c)
    a = a + 1
    return (a, b, c, temp)
```

3. (6 points) Given two arguments, `value` (a number) and `items` (a list of numbers), `numItemsSmallerThanValue` should return the number of items that are smaller than `value`. Complete the function with three simple expressions (at the places marked with `?`). The function must be recursive; only very simple arithmetic operations and recursive function calls are allowed - you may not use loops or built-in “counting” methods.

   ```python
def numItemsSmallerThanValues(value, items):
    if items == []:
        return ?
    elif ?:
        return ?
    else:
        return ?