# Quiz

1. **So Many Options...**

   (a) Implement the following function `partition_options` which outputs all the ways to partition a number `total` using numbers no larger than `biggest`.

```
def partition_options(total, biggest):
    """
    >>> partition_options(2, 2)
    [[2], [1, 1]]
    >>> partition_options(3, 3)
    [[3], [2, 1], [1, 1, 1]]
    >>> partition_options(4, 3)
    [[3, 1], [2, 2], [2, 1, 1], [1, 1, 1, 1]]
    """
    if ____________________________________________________:

        return ___________________________________

    elif ____________________________________________________:

        return ___________________________________

    else:

        with_biggest = ____________________________________________

        without_biggest = _________________________________________

        ____________ = [[_________]_______________________________]

        return with_biggest + without_biggest
```

   (b) Return the minimum number of elements from the list that need to be summed in order to add up to `T`. The same element can be used multiple times in the sum. For example, for `T = 11` and $lst = [5, 4, 1]$ we should return 3 because at minimum we need to add 3 numbers together (5, 5, and 1). You can assume that there always exists a linear combination of the elements in `lst` that equals `T`.

```
def min_elements(T, lst):
    """
    >>> min_elements(10, [4, 2, 1]) # 4 + 4 + 2
    3
    >>> min_elements(12, [9, 4, 1]) # 4 + 4 + 4
    3
    >>> min_elements(0, [1, 2, 3])
    0
    """

    if ____________________________________________________:

        return ___________________________________

    return ____________________________________________________________________________
```

(c) Reminder: don't forget to check your quiz solutions @ cs61a.org Quiz solutions can be found on the last page of the discussion solutions, which are posted at the end of each week. If you do not know where to find discussion solutions @ cs61a.org, see links.cs61a.org/quiz-sols-location