Question 1:

Provide a short description of what the following function does. This should ***not*** be a detailed description of how the code operates but rather a description that you would put in the documentation of the function to let another programmer know what the function would be used for.

```
def mystery(list):
      if len(list) == 0:
            return 0
      return list[0] + mystery(list[1:])
```

Question 2:

For the function in question 1, describe ***how*** the recursive call makes progress toward the base case. In other words, explain how it makes the problem smaller in the recursive call.

Question 3:

The following function ***should*** take as input a string and return the reversed string; however, it contains a bug. Identify the bug ***and*** modify the code to fix the bug.

```
def reverse(string):
     if len(string) == 0 or len(string) == 1:
          return string
     else:
          return string[0] + reverse(string[1:])
```

Question 4:

Write a ***recursive*** function called `count_7` that takes as input an integer and returns the number of sevens that appear in the number.

```
count_7(7171) -> 2
count_7(123) -> 0
count_7(171) -> 1
```

Question 5:

Write a function called `is_valid` that takes as input a number and behaves as follows:

- returns `True` if it is between 1 and 5 (inclusive)
- returns `False` if it is between 6 and 10 (inclusive)
- raises a `ValueError` if the number is less than 1 or larger than 10
- raises a `TypeError` if the parameter is not of type int

Question 6:

Write a function called `count_passing` that takes as input a dictionary that maps a string (student name) to an int (student score). Return the number of students who have a passing score of 70 or above.

```
scores = {
      'Parisima': 70,
      'Kimiko': 45,
      'Igor': 99,
      'Bert': 82,
      'Joanna': 77,
}
count_passing(scores) # This example would return 4
```

Question 7:

```
class Bill:

      def __init__(self, pre_tax, tax, tip_percentage):
            self.pre_tax = pre_tax
            self.tax = tax
            self.tip = pre_tax * tip_percentage
            self.total = pre_tax + tax + self.tip

      def change_tip(self, tip_percentage):
            self.tip = self.pre_tax * tip_percentage
            self.total = self.pre_tax + self.tax + self.tip

      def __str__(self):
            result = f'Pre-tax Total: {self.pre_tax}'
            result += '\n'
            result += f'Tax: {self.tax}'
            result += '\n'
            result += f'Tip: {self.tip}'
            result += '\n'
            result += f'Total: {self.total}'
            return result
```

The `Bill` class above may be used to calculate a total bill given a pre-tax amount, a tax amount, and a tip percentage. The `change_tip` method allows modifying the amount of tip given.

Write a code fragment that will

1. Create an instance of class `Bill` with pre-tax amount 100, tax amount 5, and tip percentage .20.
2. Use the `__str__` method to print the contents of the object
3. Change the tip percentage to .22.
4. Use the `__str__` method to print the updated contents of the object

Question 8:

Write a ***recursive*** function called `no_evens` that takes as input a list of integers and returns a new list that contains only the odd numbers in the list. For full credit, the items in the resulting list must be in the same order as they were in the original list. You may only access list[0], take a slice of the list, and use the len function.