Hugh O'Reilly
hjo2@rice.edu

READ ME.pdf

General:

- I used Python to complete the coding challenges.

1A Bucket Challenge:

- I decided to solve this problem using recursion, because the recursive solution was very concise and intuitive. However, this does have its drawbacks—you will need to be wary of the target values you input into this file, as large values will exceed the maximum recursion depth in Python.
- You can pass arguments at the call line.
  The input array of bucket values should be a string of integers separated by commas (e.g. 5,4,6 or 23,56,78 or 1,2,3,4,5,6 or …). In other words, there should be no spaces separating values in the array.
  The input target value should simply be an integer (e.g. 4 or 7 or 321 or …).
  For example, the following was displayed in my terminal:

```
Hughs-MBP-5:pythonPractice hughoreilly$ python ./bucket_challenge.py 1,2,3,4 413

buckets: [1, 2, 3, 4]
target value: 413
result: 1

Hughs-MBP-5:pythonPractice hughoreilly$ python ./bucket_challenge.py 1,2,3,4 456

buckets: [1, 2, 3, 4]
target value: 456
result: 1
```

1B Number to String Challenge:

- I approached this challenge with patterns in mind. When I want to memorize a long number, I first look for patterns in the digits. Maybe a digit is repeated several times, maybe there is an arithmetic sequence, maybe two adjacent digits multiply to be the digit to the right of them. Unconventional approaches, like looking for words or codes in the digits, rarely help me—in fact, those approaches are often more confusing than anything else. So I ignore them, and if I can't find patterns to represent all of the digits in a number, I simply divide those digits into manageable chunks and memorize them together. This is precisely what my file *num_to_word.py* attempts to accomplish.
- I left plenty of comments and docstrings in the file, so I hope you consider that to be sufficient documentation.
- You can pass arguments at the call line (i.e. you can pass a *single* number at the call line).
  The input number should be a 10-20 digit integer.
  For example, the following was displayed in my terminal:

```
Hughs-MBP-5:pythonPractice hughoreilly$ python ./num_to_word.py 1234567898765432
1

Input: 12345678987654321
Output: 1234 (year) - 56789 (arithmetic) - 87654321 (arithmetic)

(arithmetic), the digits form an arithmetic sequence
(geometric), the digits form a geometric sequence
(year), the digits represent a year after 1199
(product), the last number is the product of the first two digits
(symmetry), the digits are symmetric
```

2 File Challenge:

- I believe my pdf document addressing this challenge is pretty thorough.