Assignment 4

revised 2/18/2025

Ryan Waterman

4/6/2025

Question 1

1. import the random library.
2. Use `random.seed(10)` to initialize a pseudorandom number generator.
3. Create a list of 50 random integers from 0 to 15. Call this list `int_list`.
4. Print the 10th and 30th elements of the list.

You will need to use list comprehension to do this. The syntax for list comprehension is: <new_list> = `[<expression> for <item> in <iterable>]`. For this question your expression will be a randint generator from the random library and your iterable will be `range()`. Researh the documentation on how to use both functions.

In [6]:

```
import random
```

In [ ]:

```
random.seed(10)

int_list = [random.randint(0,15) for i in range(50)]

print(int_list[9], int_list[29])
```

```
[1, 13, 15, 0, 6, 14, 15, 8, 5, 1, 15, 10, 2, 7, 11, 1, 13, 4, 11, 12, 13, 9, 8, 14,
5, 9, 11, 4, 14, 7, 14, 12, 1, 0, 7, 4, 6, 9, 11, 7, 10, 14, 13, 15, 2, 10, 5, 7, 1
3, 7]
1 7
```

Question 2

1. import the string library.
2. Create the string `az_upper` using `string.ascii_uppercase`. This is a single string of uppercase letters
3. Create a list of each individual letter from the string. To do this you will need to iterate over the string and append each letter to the an empty list. Call this list `az_list`.
4. Print the list.

You will need to use a for-loop for this. The syntax for this for-loop should be:

`for i in string>:

`

In [15]:

```
import string

# First, doing this the way the question asks
az_upper = string.ascii_uppercase

az_upper_list = []
for i in az_upper:
    az_upper_list.append(i)

print(az_upper_list)

# I prefer to do this operation with list comprehension, though
az_upper = [letter for letter in string.ascii_uppercase]

print(az_upper)
```

```
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
```

Question 3

1. Create a set from 1 to 5. Call this `set_1`.
2. Create a set from int_list. Call this `set_2`.
3. Create a set by finding the `symmetric_difference()` of `set_1` and `set_2`. Call this `set_3`.
4. What is the length of all three sets?

In [19]:

```
set_1 = set(range(1, 6))
set_2 = set(int_list)
set_3 = set.symmetric_difference(set_1, set_2)

print(set_1, set_2, set_3)
print(len(set_1), len(set_2), len(set_3))
```

```
{1, 2, 3, 4, 5} {0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15} {0, 3, 6, 7, 8,
9, 10, 11, 12, 13, 14, 15}
5 15 12
```

Question 4

Complete exercise 9.15.3 from Think Python by Downey

https://allendowney.github.io/ThinkPython/chap09.html

In [23]:

```
def is_palindrome(word):
    return True if word==''.join(reversed(word)) else False
```

In [31]:

```
word_list = [    #Courtesy of ChatGPT
    "racecar",   # Palindrome
    "table",     # Non-palindrome
    "rotor",     # Palindrome
    "lamp",      # Non-palindrome
    "repaper",   # Palindrome
    "level",     # Palindrome
    "mirror",    # Non-palindrome
    "kayak",     # Palindrome
    "river",     # Non-palindrome
    "civic",     # Palindrome
    "phone",     # Non-palindrome
    "rotator",   # Palindrome
    "paper",     # Non-palindrome
    "stats",     # Palindrome
    "chair",     # Non-palindrome
    "madam",     # Palindrome
    "camera",    # Non-palindrome
    "refer",     # Palindrome
    "gadget",    # Non-palindrome
    "noon",      # Palindrome
    "window",    # Non-palindrome
    "mom",       # Palindrome
    "textbook",  # Non-palindrome
    "pop",       # Palindrome
    "random"     # Non-palindrome
]

for word in word_list:
    if len(word) >= 7 and is_palindrome(word):
        print(word)
```

```
racecar
repaper
rotator
```