Assignment 4

revised 2/18/2025

Evelyn Barragan

01/29/2026

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 [39]:
```python
# import random
import random

# initialize a pseudorandom number generator
random.seed(10)

# use list comprehension
int_list = [random.randint(0,15) for _ in range(50)]

print(int_list)

print("The 10th element of the list is: ", int_list[9])
print("The 30th element of the list is: ", 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, 1
5, 2, 10, 5, 7, 13, 7]
The 10th element of the list is:  1
The 30th element of the list is:  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>:    <list operation>`

In [40]:
```
# import string
import string

# create string az_upper
az_upper = string.ascii_uppercase

# initialize az_list
az_list = []

# iterate over string and append each letter
for letter in az_upper:
    az_list.append(letter)

print(az_list)
```

```
['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 [41]:
```
# Create set_1
set_1 = {1, 2, 3, 4, 5}

# Create set_2
set_2 = set(int_list)

# find symmetric difference
set_3 = set_1.symmetric_difference(set_2)
```

```
# find lengths of sets
print("Length of set 1 is:", len(set_1))
print("Length of set 2 is:", len(set_2))
print("Length of set 3 is:", len(set_3))
```

```
Length of set 1 is: 5
Length of set 2 is: 15
Length of set 3 is: 12
```

Question 4

Complete exercise 9.15.3 from Think Python by Downey

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

Python provides a built-in function called reversed that takes as an argument a sequence of elements – like a list or string – and returns a reversed object that contains the elements in reverse order.

In [42]:
```
reversed('parrot')
```

Out[42]:
```
<reversed at 0x13f3b25c0>
```

If you want the reversed elements in a list, you can use the list function.

In [43]:
```
list(reversed('parrot'))
```

Out[43]:
```
['t', 'o', 'r', 'r', 'a', 'p']
```

Or if you want them in a string, you can use the join method.

In [44]:
```
''.join(reversed('parrot'))
```

Out[44]:
```
'torrap'
```

So we can write a function that reverses a word like this.

In [45]:
```
def reverse_word(word):
    return ''.join(reversed(word))
```

A palindrome is a word that is spelled the same backward and forward, like “noon” and “rotator”. Write a function called is_palindrome that takes a string argument and returns True if it is a palindrome and False otherwise.

In [49]:
```
# your code here
def is_palindrome(word):
    return word == reverse_word(word)
```

You can use the following loop to find all of the palindromes in the word list with at least 7 letters.

In [50]:
```
word_list = ['apple', 'rotator', 'airport', 'racecar', 'noon']
for word in word_list:
    if len(word) >= 7 and is_palindrome(word):
        print(word)
```

```
rotator
racecar
```