CIS 210 Fall 2018
Lab Weeks 8 and 9

-- assert and try/except
-- practice with dictionaries and lists and finding and fixing bugs

# Exercise: assert

# Write a function, double_your_num, that asks the user
# to enter an integer between 1 and 10, and then
# prints the number * 2.
# Check that the input value is valid (type and range).
# Test your code with correct and incorrect input values.

def double_your_num():
    #==========================================================

# Exercises: try/except

# Given the following Python code
# to increment the first n integers
# in a list of integers by one:

def incr_li(li, n):
    '''(list of ints, int) -> None

    Increment the first n values in
    a list by 1 and print the new list.

    >>> incr_li([1, 2, 3, 4], 2
    [2, 3, 3, 4]
    >>> incr_li([1, 2, 3, 4], 5
    [2, 3, 4, 5]
    ...'''
    newli = li[:]
    # slicing the entire list creates a copy
    for i in range(n):
        newli[i] += 1
    print(newli)

    return None
# Revise the function to include
# try/except to catch the issue with
# the second example in the docstring.

#==================================
# Generate a ZeroDivisionError:
# Write Python code to catch a zero
# division error and print a msg:
# Denominator cannot be 0.

#==================================
# Generate a KeyError:
# Write Python code to catch a key
# error and print a message:
# <incorrect value> is not a key.

#==================================
# Generate a FileNotFoundError:
# Write Python code to catch a
# file not found error and print
# a message:
# Where is <filename>??

#==================================
# Revise double_your_num:
# for an invalid input value, print an
# error message and ask the user to re-enter
# the number, until a valid value is entered.
Exercises: Dictionaries

(1) Given the following lists of dates and temperatures:

```python
days = ['Mo', 'Tu', 'We', 'Th']
tems = [55, 23, 42, 44]
```

Write a function, `createTempD`, that takes the two lists and returns a dictionary with the days as keys and the temperatures as the values. Assign the result of executing `createTempD` to `dd`, which will be used in the exercises that follow.

(2) Use `dd` to find the temperature for 'We'.

(3) Add a temperature of 32 for 'Fr'.

(4) Create a sorted list of all of the temperatures in `dd`.

(5) Add a temperature of 60 for 'Saturday'.

(6) Delete the temperature for 'Saturday' and add a temperature of 60 for 'Sa'.

(7) Add a temperature of Saturday's temperature plus 10 degrees for 'Su'.
EXERCISES: Lists

Although Python provides us with many list methods, it is good practice and very instructive to think about how they are implemented. Implement a Python function that works like the following list methods (of course, don’t use the Python methods in your implementation):

(a) count

(b) Does the `my_in` function implement Python in i.e., return `True` if an item is in the list, and `False` otherwise?

```python
def my_in(li, i):
    for item in li:
        if item == i:
            return True
    else:
        return False
```

(c) index – return -1 if not in the list

(d) reverse