Quiz #4 Review

Given this dictionary:

```
states = {'AL': ['Alabama', 'Montgomery'],
          'CO': ['Colorado', 'Denver'],
          'PA': ['Pennsylvania', 'Harrisburg'] }
```

What code will add "UT" as key with the associated value of ['Utah', 'Salt Lake City'] to the dictionary?

A. states['UT'] = ['Utah', 'Salt Lake City']

What will be the following code do?

```
fname = 'somefile.txt'
f = open(fname)
x = f.read()
print(x)
```

A:
opens somefile.txt, reads its contents into variable x, then prints x

---

```
states = {'AL': ['Alabama', 'Montgomery'],
          'CO': ['Colorado', 'Denver'],
          'PA': ['Pennsylvania', 'Harrisburg'] }
```

What will the following code display?

```
for s in states:
    print(s)
```

A. All the keys, or "AL", "CO", "PA"

```
def addNums (w, x, y = 10, z = 20):
    sum = w + x + y + z
    if sum > 100:
        return 'huge'
    if sum%2 == 0:
        return 'even'
    return 'odd'
    print ('End of function')

print (addNums( ))
```

A. Illegal – not enough parameters

Show the code that will create a dictionary called "presidents" with (1) a key of integer 1 and the associated value of "Washington", (2) a key of 2 with value "Adams", and a key of 3 with value "Jefferson".

A. presidents = {1: "Washington", 2: "Adams", 3: "Jefferson"}

What will be the display output of the following code?

```
def addNums (x, y, z):
    return x + y + z

addNums(2, 3, 5)
```

A. Nothing will be displayed/printed

```
def addNums (w, x, y = 10):
    sum = w + x + y + z
    if sum > 100:
        return sum, 'huge'
    if sum%2 == 0:
        return sum, 'even'
    return sum, 'odd'

z = 20
a, b = addNums(3, 5, 7)
print (a, b)
```

A. 35, odd

```
def addNums (inNums):
    sum = 0
    for n in inNums:
        sum += n
    return sum

numList = [1, 3, 5, 7, 9]
print (addNums(numList))
```

A. 25

```
def printGreek (items):
    for n in range (len(items)):
        print(n + 1, items[n])
    return True

items = ['alpha', 'beta', 'gamma', 'delta']
printGreek (items)
```

A:
1 alpha
2 beta
3 gamma
4 delta

Write the code needed to create a function called 'capital' that accepts two parameters: (1) a dictionary of U.S. states with their two letter abbreviations as key and names and capitals as a value list, and (2) the abbreviation of a state. The function returns the name of that state's capital or 'not found' if the abbreviation is not in the dictionary. The dictionary looks like this:

states = {'AL': ['Alabama', 'Montgomery', …. 'WY': ['Wyoming', 'Cheyenne'] }

A.

```
def capital (indict, abbrev):
    if abbrev in indict:
        return indict[abbrev] [1]
    else:
        return 'not found'
```

```
def addNums (w, x, y = 10, z = 20):
    sum = w + x + y + z
    if sum > 100:
        return 'huge'
    if sum%2 == 0:
        return 'even'
    return 'odd'
    print ('End of function')

print (addNums(3, 5, 7))
```

Dictionaries are collections of data elements:
- non-sequential (unordered)
- Mutable
- ***can access by iteration, but <u>not</u> using an index***

Lists are sequential (ordered)
- can retrieve by a relative index ([0], [1], [2],…)

Dictionaries are **key-value pairs**
- can retrieve a value using its key
- the key "maps" to the value
- can iterate through a dictionary – by key, but not by index

Keys are immutable and unique

Values can be numbers, strings, lists, tuples, or other dictionaries

A. odd

- Example:
  states = {"AZ" : "Arizona", "FL" : "Florida", "ME" : "Maine"}

To add Montana:	states['MT'] = ['Montana']
Check membership:	if 'MT' in states: …
Delete an entry pair:	del states['ME']
Retrieve all keys:	list (states.keys())
Retrieve all key-values:	list(states.items())
Pop an item:	states.pop('CA')
Create a copy:	st_copy = states.copy()

Iterate by keys:	for s in states: # print key & value
	print(s, states[s])