Computer Science 125
1 October 2020

1. Why would we use a dictionary in Python?

- Stores key-value pairs —
  If we want to look up information based on some identifier (name, word, ID number), then use a dictionary.

Example: store student information by student ID number
ID: ________________

- Add/Remove values by identifiers
- Fast lookup

2. What is the output of the following code?

Create a dictionary

key value key value

```
D = {"milk":"milch", "coffee":"kaffee", "water":"wasser"}
print( D["coffee"] )
print( D.keys() )
```

key value

— prints "kaffee"

— prints list of keys

3. How could we add another key-value pair to the dictionary above?

D["juice"] = "saft"

4. How can we print out all of the keys and values in the dictionary above?

print(D)

print(D.keys())

print(D.values())

print(D.items())

5. **Write a function that uses a dictionary to count the frequencies of letters in a string. Print the letters and counts in alphabetical order.**

```
def countLetters(text):

    counts = {}

    for char in text:

        if char in counts:    # dictionary already has this key
            counts[char] = counts[char] + 1
        else:    # dictionary does not have this key yet
            counts[char] = 1    # stores count of 1 for this char.

    return counts
```

Practice with Dictionaries – SOLUTIONS

Working with a partner/group, use the following steps to solve each of the following problems.

(a) Plan your code on the white board (either on the classroom wall or on Zoom). Write out your entire program. Think about what errors might occur and how to fix them.

(b) Plan multiple test cases. What input will you send to your function? For each input, what value should be returned?

(c) *Only after you have completed steps (a) and (b) should you type your code in Python.*

(d) After you have typed your code, run your test cases. Does your code work? If not, how can you fix it?

1. In some dictionaries, the values are all integers. Write a function `allIntVals(d)` that accepts a dictionary `d` as a parameter and checks to see whether all of the values in the dictionary are integers. If they are, your function returns `True`; otherwise it returns `False`.

```
def allIntVals(d):
    for v in d.values():
        if not type(v) is int:
            return False
    return True
```

2. Write a function `sumValues(d)` that accepts a dictionary as a parameter. If dictionary `d` contains only integer values (call your function `allIntVals(d)`), then return the sum of the values. Otherwise, print an error message.

```
def sumValues(d):
    if not allIntVals(d):
        print("Error: the dictionary contains non-integer value")
    else:
        total = 0
        for v in d.values():
            total += v
    return total
```

3. Write a function `merge(d1, d2)` that merges two dictionaries and returns the result. Any key that appears in `d1` or `d2` should appear in the returned dictionary. If a key appears in only one of `d1` or `d2`, then its value in the returned dictionary should be the same as its value in `d1` or `d2`. However, if a key appears in both `d1` and `d2`, then its value in the returned dictionary should be the sum of its values in `d1` and `d2`.

   For example, if `d1 = {'a':5, 'b':2}` and `d2 = {'a':3, 'c':4}`, then the returned dictionary should be `{'a':8, 'b':2, 'c':4}`.

Make sure your function does not modify either or the dictionaries sent as parameters! You might want to use the dictionary `.copy()` method to make a copy of a dictionary.

```
def merge(d1, d2):
    nd = d1.copy()
    for k in d2:
        if k in nd:
            nd[k] += d2[k]
        else:
            nd[k] = d2[k]
    return nd
```

4. **Bonus:** Write a function `wordfrequencies(text)` that accepts a string of text and uses a dictionary to count the frequencies of each word in the text. Remove non-alphabetic characters from the text before counting words. Then modify your code so that you can read in a (possibly large) text file and compute the frequencies of all words in the file.

```
import string
def wordFrequencies(text):
    wordCounts = {} #new dictionary
    words = text.split()
    for w in words:
        alpha = ""
        for c in w:
            if c in string.ascii_letters:
                alpha += c.lower()
        #now check whether alpha is already in the dictionary
        if alpha in wordCounts:
            wordCounts[alpha] += 1
        else:
            wordCounts[alpha] = 1
    return wordCounts
```