

Name: (as it would appear on official course roster)
UCSB email address: _____ @ucsb.edu
Lab Section:
Optional: name you wish to be called if different from above
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")

h07: Dictionaries

Assigned: Tuesday, May 28th, 2019

Due: Tuesday, June 4th, 2019

Points: 100

- You may collaborate on this homework with AT MOST one person, an optional "homework buddy". MAY ONLY BE TURNED IN THE LECTURE LISTED ABOVE AS THE DUE DATE. There is NO MAKEUP for missed assignments; in place of that, we drop the single lowest score (if you a zero, that is the lowest score.)
- IMPORTANT:** When submitting this homework:
 - DO NOT USE STAPLES
 - WRITE YOUR NAME ON EACH PAGE IN THE SPACE PROVIDED
 - USE DARK INK PENS – PLEASE DO NOT USE PENCIL
 - PRINT THIS HOMEWORK DOUBLE-SIDED PLEASE!

<ul style="list-style-type: none"> REMEMBER: If you use code/techniques we have not learned in class, you will NOT get credit!

READING ASSIGNMENT: Read Chapter 6.1 in Perkovic, review your lecture slides/notes. Then complete these problems.

1. (35 pts) Below is a transcript of a shell session in Python. Fill in what would be printed by the shell after each set of statements. [Hint: TRY each one in Python.]

a. (2 pts)

```
>>> myDict = {"Mei":95, "Bob":85, "Jose":93, "Diana":100}
>>> myDict["Jose"]
```

b. (3 pts)

```
>>> for item in myDict:
... print(item)
```

c. (5 pts)

```
>>> myDict["Raj"] = 87
>>> names = list(myDict.keys())
>>> names.sort()
>>> names
```

Name:

(as it would appear on official course roster)
--

d. (5 pts)

```
>>> for name in names:  
... print(myDict[name])
```

e. (5 pts)

```
>>> for name in names:  
... print(name,": ",myDict[name])
```

f. (10 pts)

```
>>> for name in sorted(myDict):  
... print(name,": ",myDict[name])
```

Name:

(as it would appear on official course roster)
--

2. (30 pts) Write a function named **week()** that takes no arguments. It will repeatedly ask the user to enter an abbreviation for a day of the week (Mo, Tu, We, Th, Fr, Sa, or Su) and then print the corresponding day. If the user enters anything other than one of those 7 abbreviations, the program ends. You **MUST** use a dictionary in your solution or you won't get any credit. If you write the definition in **7 lines total, or fewer, you get 10 extra credit points!**

An example run would look like this:

```
Enter weekday abbreviation: Tu
```

```
Tuesday
```

```
Enter weekday abbreviation: Sa
```

```
Saturday
```

```
Enter weekday abbreviation:
```

(there's nothing that's printed out at the end – the program just quits)

Name:

(as it would appear on official course roster)

3. (35 pts) Write a function named **printSorted()** that takes a dictionary as its only input parameter, and it prints out the key/value pairs in order by key.

```
def printSorted(myDict):
```