

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

1

h04

CS8 M18

h04: Perkovic 6.1-6.2 (Dictionaries, Sets), Perkovic 6.3-6.4 (Encodings, random), Review (tracing functions), Number Conversions

ready?	assigned	due	points
true	Tue 08/28 09:30AM	Tue 09/04 09:30AM	386

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE.

There is NO MAKEUP for missed assignments, and you may not submit work in advance, or on behalf of another person.

In place of that, we drop the lowest score (if you have a zero, that is the lowest score.)

READING ASSIGNMENT

Please read Perkovic 6.1-6.2 (Dictionaries, Sets). Then complete these problems.

- (10 pts) Please fill in the information at the top of this homework sheet, including your name and umail address. If the other two items apply, please fill them in as well. Please do this every single time you submit homework for this class. It is important to fill in both name and umail every time, since handwriting is sometimes difficult to decipher. Having both helps us ensure you get credit for your work.

Also: JUST write your name at the top of all sheets and turn in all sheets UNCONNECTED.

DO NOT staple, paper clip, spit-fold-and-tear, or do ANYTHING that would make it difficult to automatically feed your paper through a scanner.

- Suppose we define the following Python variable:

```
capitols = { "CA": "Sacramento", "NV": "Carson City",
             "AZ": "Phoenix", "WA": "Seattle", "OR": "Portland" }
```

What would the value of each of the following Python expressions be? Fill in the blanks. Remember that `type` returns types in the format `<class 'int'>`, `<class 'list'>`, `<class 'str'>`, etc. Use the correct value for full credit.

Pts	Expresssion	Value
(5 pts)	<code>type(capitols)</code>	
(5 pts)	<code>capitols["CA"]</code>	
(5 pts)	<code>type(capitols["CA"])</code>	

Pts	Expresssion	Value
(5 pts)	<code>len(capitols["OR"])</code>	
(5 pts)	<code>capitols["OR"][4:8]</code>	

3. Suppose we define states as follows:

```
states = {
    "AZ" : { "capitol" : "Phoenix", "borders" : {"AZ","NV"} },
    "CA" : { "capitol" : "Sacramento", "borders" : {"AZ","NV","OR"} },
    "NV" : { "capitol" : "Carson City", "borders" : {"OR","AZ","CA"} },
    "OR" : { "capitol" : "Portland", "borders" : {"WA","CA","NV"} },
    "WA" : { "capitol" : "Seattle", "borders" : {"OR"} }
}
```

What would the value of each of the following Python expressions be? Fill in the blanks. Remember that `type` returns types in the format `<class 'int'>`, `<class 'list'>`, `<class 'str'>`, etc. Use the correct value for full credit.

a.

Pts	Expression	Value
(5 pts)	<code>type(states)</code>	
(5 pts)	<code>type(states["OR"])</code>	
(5 pts)	<code>type(states["OR"] ["capitol"])</code>	
(5 pts)	<code>type(states["CA"] ["borders"])</code>	

b.

Pts	Expression	Value
(5 pts)	<code>len(states.keys())</code>	
(5 pts)	<code>len(states["NV"].keys())</code>	
(5 pts)	<code>len(states["OR"]["capitol"])</code>	
(5 pts)	<code>states["OR"]["capitol"]</code>	
(5 pts)	<code>states["WA"]["borders"]</code>	

c.

Pts	Expression	Value
(5 pts)	<code>"CA" in states["WA"]["borders"]</code>	
(5 pts)	<code>"NV" in states["OR"]["borders"]</code>	
(5 pts)	<code>"AZ" in states["CA"]["borders"]</code>	
(5 pts)	<code>states["WA"]["borders"] < states["CA"] ["borders"]</code>	

2

h04

CS8 M18

READING ASSIGNMENT Please read Perkovic 6.3-6.4 (Encodings, random). Then complete these problems.

4. Section 6.3 discusses character encodings, a topic that usually arises in the context of dealing with strings, and files. Two encodings are discussed: ASCII and Unicode. The text also discusses various ways of representing Unicode characters including `utf-8`, `utf-16`, and `utf-32`.

a. (10 pts) The original 7-bit ASCII character set has 128 different characters, numbered 0 through 127, but not all of them are “printable” characters. According to our text, how many of the ASCII characters with codes 0 through 127 are considered “printable”?

b. (10 pts) Fill in the blanks in the following sequence of Python code so that the code prints out all of the lowercase characters ‘a’ through ‘z’:

```
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
```

Your answer should only involve filling in the blanks; not modifying the rest of the code in anyway. As a reminder, `end=' '` on a function call to `print` makes the print NOT go to a new line; instead, it just prints a space after each call to `print` (See Section 4.2, p. 99 in the textbook.)

```
for i in range ( _____ , _____ ) :  
    print(chr(i),end=' ')
```

3
h04
CS8 M18

4

h04

CS8 M18

5. Indicate what the output of the code will be:

<pre>for c in "UCSB": print(ord(c),sep=",")</pre>	(10 pts)
<pre>print ('\u0048\u0049\u0021')</pre>	(10 pts)

6. (10 pts) Write a Python expression that returns a random integer between 5 and 10.

7. Suppose the python variable `ucList` is defined as follows:

```
ucList=[ "UCB", "UCD", "UCI", "UCLA", "UCM", "UCR", "UCSB", "UCSC", "UCSD" ]
```

- (10 pts) Write a Python expression that returns the name of a random UC from this list (as a string). For full credit, use the most straightforward way (that way doesn't involve calculating the length of the list.)
- (10 pts) Write a Python expression that returns a pair of random UCs from this list (i.e. a list of length 2 that has two randomly selected UCs from the list. Do it in the most straightforward way possible. (If you read the whole section carefully, you'll discover this can be done in one line of code.)
- (10 pts) Section 6.3 explains the idea of *lexicographic order*. As it turns out, the list `ucList` is already in lexicographic order, though if it were not, we could put it in lexicographic order by writing `ucList.sort()`.

What is lexicographic order?

- d. (20 pts) Suppose that someone proposed that instead of putting the UCs in alphabetical order, to randomize the order (so that “UCB” wouldn’t always be first, and “UCSD” wouldn’t always be last.)

Write some Python code that would print the UC abbreviations, one per line, in a random order. Use the most straightforward way of doing it.

5

h04

CS8 M18

The problems below are all based on readings already completed. If you need to, review the appropriate sections in the textbook.

8. For each of the following, there is a Python function definition in the box at right. Assume that it has been loaded into `idle3` and that we've selected `Run Module` (or pressed `F5`.) Then we typed in the function call shown, and something is printed as a result. Which of the answers shown matches what is printed? (Multiple choice)

6

h04

CS8 M18

<p>(a) (10 pts)</p> <pre>>>> ===== RESTART ===== >>> >>> mystery2([3,4,5,6,7]) >>></pre> <p>a. 4 b. 7 c. [4, 6] d. [7] e. Error:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"><pre>... result = result + x TypeError: can only concatenate list (not "int") to list</pre></div> <p>f. None of the above</p>	<pre>def mystery2(aList): """ Computes something from list. What? """ result = [] for x in aList: if (x % 2 == 0): result = result + [x] return x</pre>
--	--

For each of the following, there is a Python function definition in the box at right. Assume that it has been loaded into `idle3` and that we've selected `Run Module` (or pressed `F5`.) Then we typed in the function call shown, and something is printed as a result. Which of the answers shown matches what is printed? (Multiple choice)

<p>(b) (20 pts)</p> <pre>>>> ===== RESTART ===== >>> >>> mystery3([10,20,25]) >>></pre> <p>a. 10 b. [10, 20] c. [25] d. 30 e. Error:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"><pre>... result = result + x TypeError: can only concatenate list (not "int") to list</pre></div> <p>f. None of the above</p>	<pre>def mystery3(aList): """ Computes something from list. What? """ result = [] for x in aList: if (x % 2 == 0): result = result + [x] return result</pre>
--	---

7

h04

CS8 M18

<p>(c) (20 pts)</p> <pre>>>> ===== RESTART ===== >>> >>> mystery4([7,14,102,9]) >>> _____</pre> <p>a. 9 b. [14, 102] c. 144 d. 0 e. Error:</p> <pre>... result = result + x TypeError: can only concatenate list (not "int") to list</pre> <p>f. None of the above</p>	<pre>def mystery4(aList): """ Computes something from list. What? """ result = [] for x in aList: if (x % 2 == 0): result = result + [x] return result</pre>
<p>(d) (20 pts)</p> <pre>>>> ===== RESTART ===== >>> >>> mystery5([7,14,102,9]) >>> _____</pre> <p>a. 9 b. [14, 102] c. 144 d. 0 e. Error:</p> <pre>... result = result + x TypeError: can only concatenate list (not "int") to list</pre> <p>f. None of the above</p>	<pre>def mystery5(aList): """ Computes something from list. What? """ result = 0 for x in aList: if (x % 2 == 0): result = result + 1 return result</pre>

8

h04

CS8 M18

(e) (20 pts)

```
>>> ===== RESTART =====  
>>>  
>>> mystery6([7,14,102,9])  
>>> _____
```

- a. 9
- b. [14, 102]
- c. 144
- d. 0
- e. Error:

```
...     result = result + x  
TypeError: can only concatenate  
list (not "int") to list
```

f. None of the above

```
def mystery6(aList):  
    """  
    Computes something from list.  What?  
    """  
  
    result = 0  
    for x in aList:  
        if (x % 2 == 0):  
            result = result + 1  
  
    return result
```

VIDEO ASSIGNMENT

If the lecture about number conversions was sufficient for you to be confident understand how to do this homework assignment, you do not need to watch the videos. Just solve the problems.

Otherwise, please consult the following page, and the videos listed:

- https://ucsb-cs8.github.io/topics/number_conversions/

Then complete these problems.

9. (8 pts) Convert 1100 0000 1100 0110 from binary to base 16

10. (8 pts) Convert 1011 0000 from binary to decimal

11. (8 pts) Convert 67 from decimal to base 2

9
h04
CS8 M18

10

h04

CS8 M18

12. (8 pts) Convert 001 111 from base 2 to octal

13. (8 pts) Convert 0010 1001 from binary to decimal

14. (8 pts) Convert 8a49 from base 16 to base 2

15. (8 pts) Convert 0001 0100 from binary to base 10

16. (8 pts) Convert 1010 1101 0101 0000 from base 2 to hexadecimal

17. (8 pts) Convert 0010 0100 from binary to base 10

18. (8 pts) Convert 110 100 000 from binary to base 8

19. (8 pts) Convert 0111 1101 1100 0100 from base 2 to hexadecimal

20. (8 pts) Convert 43 from octal to base 2