

Test 1: Compsci 101

Kristin Stephens-Martinez

February 18, 2020

Name: _____ (1/2 pt)

NetID/Login: _____ (1/2 pt)

Section Number: _____

Honor code acknowledgment (signature) _____ (1 pt)

	Value	Grade
Front Page	2 pts.	
Problem 1	34 pts.	
Problem 2	17 pts.	
Problem 3	6 pts.	
Problem 4	7 pts.	
Problem 5	24 pts.	
TOTAL:	90 pts.	

Bubble sheet: Tear off the bubble sheet from the front of the exam and fill in the Name, ID, and Date field. Your ID is your NetID.

This test has 14 pages be sure your test has them all. Do NOT spend too much time on one question — remember that this class lasts 75 minutes.

In writing code you do not need to worry about specifying the proper `import statements`. Don't worry about getting function or method names exactly right. Assume that all libraries and packages we've discussed are imported in any code you write.

Be sure your name and NetID are legible on this page and that your NetID appears at the top of every page.

There are two blank pages at the end of the test for extra work space.

PROBLEM 1 : (Name/Type/Value (34 points))**Part A (28 points)**

Consider the following variables and their values in answering the questions below. For each statement, provide the type and value after the statement is executed. Up to two of these questions will have the answer "None of the Above".

```
insects = ['grasshopper', 'cockroach', 'fire ant', 'water strider',  
          'praying mantis', 'water boatman', 'ant', 'silkworm']
```

```
say = 'I find that the harder I work, the more luck I seem to have.'
```

```
insects[-2][1]
```

1. What is the type? (a) list (b) string (c) Error (d) None of the Above
 2. What is the value? (a) 'ant' (b) 'n' (c) ['a'] (d) Error (e) None of the Above
-

```
len(say.split("the"))
```

3. What is the type? (a) float (b) int (c) Error (d) None of the Above
 4. What is the value? (a) 2.0 (b) 3 (c) 4 (d) Error (e) None of the Above
-

```
10//4
```

5. What is the type? (a) float (b) int (c) Error (d) None of the Above
 6. What is the value? (a) 2 (b) 2.0 (c) 2.5 (d) Error (e) None of the Above
-

```
insects[2]
```

7. What is the type? (a) list (b) string (c) Error (d) None of the Above
 8. What is the value?
 - (a) 'fire ant'
 - (b) 'water strider'
 - (c) ['fire ant']
 - (d) Error
 - (e) None of the Above
-

```
5 > 6
```

9. What is the type? (a) boolean (b) int (c) Error (d) None of the Above
 10. What is the value? (a) 5 (b) False (c) True (d) Error (e) None of the Above
-

```
"cockroach" in insects
```

11. What is the type? (a) boolean (b) int (c) Error (d) None (e) None of the Above
 12. What is the value? (a) 2 (b) False (c) True (d) Error (e) None of the Above
-

```
10-7
```

13. What is the type? (a) float (b) int (c) Error (d) None of the Above
 14. What is the value? (a) 17 (b) 3 (c) 3.0 (d) Error (e) None of the Above
-

```
11/2
```

15. What is the type? (a) float (b) int (c) Error (d) None of the Above
 16. What is the value? (a) 5 (b) 5.5 (c) 6 (d) Error (e) None of the Above
-

```
"fire" in insects
```

17. What is the type? (a) boolean (b) int (c) Error (d) None (e) None of the Above
 18. What is the value? (a) 4 (b) False (c) True (d) Error (e) None of the Above
-

`insects[5:]`

19. What is the type? (a) `list` (b) `string` (c) `Error` (d) None of the Above
20. What is the value?
- (a) `'water boatman'`
- (b) `['praying mantis', 'water boatman', 'ant', 'silkworm']`
- (c) `['praying mantis', 'water boatman', 'ant']`
- (d) `Error`
- (e) None of the Above
-

`say[2:6]`

21. What is the type? (a) `int` (b) `string` (c) `Error` (d) None of the Above
22. What is the value? (a) `' fin'` (b) `'find'` (c) `4` (d) `Error` (e) None of the Above
-

`"h" in say`

23. What is the type? (a) `boolean` (b) `int` (c) `Error` (d) None of the Above
24. What is the value? (a) `8` (b) `False` (c) `True` (d) `Error` (e) None of the Above
-

`(not (2 == 2)) or (8 >= 8)`

25. What is the type? (a) `boolean` (b) `int` (c) `Error` (d) None of the Above
26. What is the value? (a) `8` (b) `False` (c) `True` (d) `Error` (e) None of the Above
-

`5 % 3`

27. What is the type? (a) `float` (b) `int` (c) `Error` (d) None of the Above
28. What is the value? (a) `1` (b) `2` (c) `2.0` (d) `3` (e) None of the Above
-

Part B (6 points)

Given the code below. Choose the output for each `print` statement. Up to one of these questions may have the answer “None of the Above”.

```
lstA = [[0, 1], 2]
lstB = [3, [4, 5]]
lstC = lstA + lstB
lstB.append('hi')
lstC[0].append('bye')
```

```
print(lstA) #29
```

29. What is printed?

- (a) `[[0, 1, 'bye'], 2]`
 - (b) `[[0, 1], 2, 'bye']`
 - (c) `[[0, 1], 2]`
 - (d) Error
 - (e) None of the Above
-

```
print(lstB) #30
```

30. What is printed?

- (a) `[3, 'hi', [4, 5]]`
 - (b) `[3, [4, 5], 'hi']`
 - (c) `[3, [4, 5]]`
 - (d) `[[3, [4, 5, 'hi']]]`
 - (e) None of the Above
-

```
print(lstC) #31
```

31. What is printed?

- (a) `[[0, 1, 'bye'], 2, 3, [4, 5], 'hi']`
 - (b) `[[0, 1, 'bye'], 2, 3, [4, 5]]`
 - (c) `[[0, 1], 2, 3, [4, 5], 'hi']`
 - (d) `[[0, 1], 2, 3, [4, 5]]`
 - (e) None of the Above
-

PROBLEM 2 : (Spot the Bug (17 points))

The following function is an implementation of the Round problem from APT Quiz 0. It rounds a number to the nearest number that is divisible by X. But this function is buggy! Below are some example calls of what the function should return.

Function call	Return value
nearestX(13, 5)	15
nearestX(12, 5)	10
nearestX(-6, 4)	-4
nearestX(0, 5)	0
nearestX(9, 3)	9
nearestX(1, 5)	0
nearestX(-3, 5)	-5
nearestX(3, 2)	4

```
def nearestX(num, x):  
    above = ((num // x) + 1) * x  
    below = (num // x) * x  
  
    if above-num <= below-num:  
        return above  
    else:  
        return below
```

Part A (6 points)

The following two questions are **checkbox questions**, a.k.a. multiple options could be the answer.

32. Of the following function calls to the buggy code, which *WILL NOT* work as expected due to the code's bug? (There may be more than one)
- (a) nearestX(13, 5) should return 15
 - (b) nearestX(12, 5) should return 10
 - (c) nearestX(-6, 4) should return -4
 - (d) nearestX(0, 5) should return 0
 - (e) None of the above
33. Of the following function calls to the buggy code, which *WILL NOT* work as expected due to the code's bug? (There may be more than one)
- (a) nearestX(9, 3) should return 9
 - (b) nearestX(1, 5) should return 0
 - (c) nearestX(-3, 5) should return -5
 - (d) nearestX(3, 2) should return 4
 - (e) None of the above

Part B (5 points)

In the first two cells below, provide the arguments for a call to `nearestX` with either your own arguments or arguments from the examples that returns a *wrong* value. In the “actual return value” cell, write your function call’s return value. If the function call causes an error, write “Error” in the cell. In the “correct return value” cell, write the value it should return.

argument: <code>num</code>	argument: <code>x</code>	actual return value	correct return value

Part C (6 points)

Here is the buggy code again. Fix it so that it always returns the correct values. You should need to change no more than 3 lines of code. If you reimplement the function, you will earn either all or none of the points for this problem.

```
def nearestX(num, x):  
  
    above = ((num // x) + 1) * x  
  
    below = (num // x) * x  
  
    if above-num <= below-num:  
  
        return above  
  
    else:  
  
        return below
```

PROBLEM 3 : (Formulas (6 points))**Part A (3 points)**

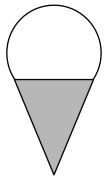
$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Above is the formula to calculate the distance between two points: (x_1, y_1) and (x_2, y_2) . Implement the function `distance` that takes the x- and y-coordinate for two points and returns the distance between them.

Assume that the `math` library has already been imported for you.

```
def distance(x1, y1, x2, y2):
    """
    x1 (int/float) - x-coordinate for first point
    y1 (int/float) - y-coordinate for first point
    x2 (int/float) - x-coordinate for second point
    y2 (int/float) - y-coordinate for second point

    Return the distance between the two points.
    """
```

Part B (3 points)

$$C = \pi r^2 \frac{h}{3}$$

$$S = \frac{4}{3} \pi r^3$$

$$I = C + \frac{3}{4} S$$

Write the function `volume` that returns the volume of an ice cream cone. Above are the formulas for the entire volume of a cone, C , and a sphere, S . The volume of the ice cream, I , is the volume of a cone with $\frac{3}{4}$ the volume of a sphere, which represents the ice cream scoop ($\frac{1}{4}$ is inside the cone and therefore redundant).

Assume that the `math` library (and therefore `math.pi`) has already been imported for you.

```
def volume(radiusCone, height, radiusScoop):
    """
    radiusCone (int/float) - the radius of the cone
    height (int/float) - the height of the cone
    radiusScoop (int/float) - the radius of the ice cream scoop

    Calculate the volume of an ice cream cone with one simple scoop of ice cream.
    """
```

PROBLEM 4 : (Cashier (7 points))

Implement the function `cashier` as described below.

Example calls are:

Function call	Return value
<code>cashier(['egg', 'pancake'])</code>	<code>[1, 1, 3.0]</code>
<code>cashier(['egg', 'pancake', 'egg and pancake'])</code>	<code>[2, 2, 5.5]</code>
<code>cashier(['egg and pancake', 'pancake', 'egg'])</code>	<code>[2, 2, 5.5]</code>
<code>cashier(['egg and pancake', 'egg', 'pancake', 'egg'])</code>	<code>[3, 2, 6.5]</code>
<code>cashier(['pancake', 'egg and pancake', 'egg', 'egg and pancake'])</code>	<code>[3, 3, 8.0]</code>

```
def cashier(orders):
    """
    orders (list of str) - A list of orders for a table at a restaurant

    ORDERS is a list of orders from a table. The restaurant only serves three
    possible orders: "egg", "pancake", and "egg and pancake" that are $1.00,
    $2.00, and $2.50 respectively.

    Return a list where the first value is the number of eggs ordered, the second is
    the number of pancakes, and the third value is the total cost for the table.

    Assume: The only strings in ORDERS are from the list of possible orders.
    """
```

PROBLEM 5 : (*Scramble (24 points)*)**Part A (5 points)**

Implement the function `swapLetters` as described below.

Example calls are:

Function call	Return value
<code>swapLetters('from', 1, 2)</code>	<code>'form'</code>
<code>swapLetters('scramble', 2, 5)</code>	<code>'scbamrle'</code>
<code>swapLetters('interest', 3, 4)</code>	<code>'intreest'</code>
<code>swapLetters('formula', 1, 5)</code>	<code>'flrmuoa'</code>

```
def swapLetters(word, first, second):
    """
    word (str) - a string with a single word in it
    first (int) - the first letter in WORD that needs to be swapped
    second (int) - the second letter in WORD that needs to be swapped

    Swap the letter at index FIRST in WORD with the letter at index SECOND of
    WORD.

    Assume: 0 < first < second < len(word)-1
    Assume: len(word) >= 4
    """
```

Part B (5 points)

Implement the function `reverseMiddle` as described below.

Example calls are:

Function call	Return value
<code>reverseMiddle('call', 1, 2)</code>	<code>'clal'</code>
<code>reverseMiddle('blank', 1, 3)</code>	<code>'bnalk'</code>
<code>reverseMiddle('string', 1, 4)</code>	<code>'snirtg'</code>
<code>reverseMiddle('combine', 3, 4)</code>	<code>'comibne'</code>
<code>reverseMiddle('scramble', 4, 6)</code>	<code>'scralbme'</code>

```
def reverseMiddle(word, first, second):
    """
    word (str) - a string with a single word in it
    first (int) - the starting letter in WORD that will be reversed
    second (int) - the ending letter in WORD that will be reversed

    Reverse all the letters in WORD between and including FIRST and SECOND.

    Assume: 0 < first < second < len(word)-1
    Assume: len(word) >= 4
    """
```

Part C (14 points)

Implement the function `scramble` as described below. `scramble` should work if its parameter `scrambler` is a function that returns a string and takes three arguments: the first is a string and the latter two are ints, like `swapLetters` and `reverseMiddle`. The ints fulfill the same constraints as `swapLetters` and `reverseMiddle` do and your code should randomly generate them. Your code should be able to use any function that fulfills this criteria, even ones that are not `swapLetters` and `reverseMiddle`. If you directly call `swapLetters` and `reverseMiddle` in your code you will lose points.

Assume that the `random` library has already been imported for you. Reminder: `random.randint(a, b)` returns a number inclusive of $[a, b]$.

Example calls are:

Function call	Return value	Notes
<code>scramble('hello world', swapLetters)</code>	<code>'hlelo wlrod'</code>	'hello' was called with the numbers 1 & 2, 'world' with 1 & 3
<code>scramble('The great and terrible oz', reverseMiddle)</code>	<code>'The gaert and terlbire oz'</code>	Only two words were changed because they are the only ones with a length ≥ 4
<code>scramble('when words fail', swapLetters)</code>	<code>'wehn wdros fial'</code>	

```
def scramble(sentence, scrambler):
    """
    sentence (str) - a sentence with multiple words that will be scrambled
    scrambler (function) - a function that takes a string and then two ints

    Scramble the words in SENTENCE by calling the function SCRAMBLER on each of the words
    in SENTENCE with len(word) >= 4. Return a string with each word in the resulting
    sentence separated by a single whitespace. For each call to SCRAMBLER, randomly pick
    two ints that fulfill the constraint: 0 < first < second < len(word)-1

    Assume: The words in SENTENCE are separated by a single whitespace and they
    are composed of only letters (i.e. no need to take into account punctuation).
    """
```

PROBLEM 6 : (*Extra Credit (1 point)*)

Predict what range your percentage grade will be on the exam. If you are correct, you will earn 1 point of extra credit on this exam. Note, that except for the last option each range excludes the first value and includes the later value. The front page has a table with the number of points for each problem.

- ☐ 95% < grade <= 100%
- ☐ 90% < grade <= 95%
- ☐ 85% < grade <= 90%
- ☐ 80% < grade <= 85%
- ☐ 75% < grade <= 80%
- ☐ 70% < grade <= 75%
- ☐ 65% < grade <= 70%
- ☐ 60% < grade <= 65%
- ☐ 55% < grade <= 60%
- ☐ 50% < grade <= 55%
- ☐ 45% < grade <= 50%
- ☐ 40% < grade <= 45%
- ☐ 35% < grade <= 40%
- ☐ 30% < grade <= 35%
- ☐ 25% < grade <= 30%
- ☐ 20% < grade <= 25%
- ☐ 15% < grade <= 20%
- ☐ 10% < grade <= 15%
- ☐ 5% < grade <= 10%
- ☐ 0% <= grade <= 5%

extra page

extrapage