Mock Exam, Week 2

Name:

Name of student to the left:

Name of student to the right:

Directions

- **This exam contains 19 questions and will last 100 minutes.**
- Two questions are long answer, while the other 17 questions have you select from multiple choices or have you write a short answer.
- Use your time wisely. If you are having too much trouble on a question, skip it and return to it later. **Avoid getting stuck.**
- In the answer options, the ↵ symbol indicates a new line. The ↵ symbol will only be used to separate lines of output and will not appear at the end of the final line.
- For questions with *circular bubbles*, you should select *exactly one* choice.
  - ○ You must choose either this option
  - ○ Or this one, but not both!
- For questions with *square checkboxes*, you may select *multiple* choices.
  - □ You could select this choice.
  - □ You could select this one too!

Staff use only.

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11
Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19			Total

1. Suppose `y = 0`. What would the following code print when run?

```
if y > 0:
    print("Good morning")
if y < 0:
    print("Good afternoon")
print("Good night")
```

- ○ `Good morning`
- ○ `Good afternoon`
- ○ `Good night`
- ○ `Good morning` ↵ `Good afternoon` ↵ `Good night`

2. For which value(s) of `a` and `b` will the following code print `JamCoders`?

```
if a and b:
    print("AddisCoder")
else:
    print("JamCoders")
```

Fill in the boxes next to all answers that print `JamCoders`.

- ☐ `a = True, b = True`
- ☐ `a = True, b = False`
- ☐ `a = False, b = True`
- ☐ `a = False, b = False`

3. For each of the following lines, fill in the blank (_____) with one of the following types: `int`, `list`, `str`, `bool`, `float`.

```
print(type("abc")) # Printed: ____________________
print(type(34)) # Printed: ____________________
print(type(True)) # Printed: ____________________
print(type(62.0)) # Printed: ____________________
print(type("False")) # Printed: ____________________
```

4. Consider the following lines of code:

```
def jah():
    return 'jah'

def ri():
    print('bah')
    return 'ri'

jah()
print(jah())
ri()
print(ri())
```

What is the printed output of the code above?

- ○ jah ↵ bah ↵ ri
- ○ bah ↵ ri
- ○ jah ↵ jah ↵ bah ri ↵ ri
- ○ jah ↵ bah ↵ bah ↵ ri

5. Suppose `L = [1, 6, 1, 4, 1, 10, 2]` and `final = 0`. What would the code output when run?

```
for i in L:
    final += i
print(final)
```

- ○ 0
- ○ 16141102
- ○ 25
- ○ 2

6. The following code searches through a list for an item and returns the **index** of its first occurrence. Fill in the blanks to complete the function.

```
def find_in_list(lst, item):
   for i in range(len(lst)):
       if __________ == item:    # YOUR CODE HERE
           return __________     # YOUR CODE HERE
   return None
```

7. Fill in the following code to correctly find the first **even** index where the element `desired` appears in the list `lst`.

For example, if `desired` has value 8 and `lst` is [7, 8, 8, 5], your function should return the index 2.

```
def find_first_even_index(lst, desired):
    for i in _________________:                    # YOUR CODE HERE
        if _____ % 2 == 0 and ________ == desired: # YOUR CODE HERE
            return ______                          # YOUR CODE HERE
```

8. What will be printed after running the following code? Write your answers on the dashed lines.

```
def func1(lst):
    return lst[0]

def func2(lst):
    print lst[0]

func1(["hi","bye"])  # Printed: ____________
func2([0,1,2])       # Printed: ____________
```

9. What is printed when you run the following code?

```
count = 0
while True:
       if count == 5:
            break
       print(count)
```

- ○ 0 ↵ 1 ↵ 2 ↵ 3 ↵ 4
- ○ 0 ↵ 0 ↵ 0 ↵ 0 ↵ 0
- ○ An error occurs, and nothing is printed.
- ○ 0 ↵ 0 ↵ 0 ↵ 0 ↵ 0 ↵ 0 ↵ 0 ↵ 0 . . . (An infinite loop that keeps printing 0 ↵)

This question has two parts.

Complete the following function to compute the *pairwise product* of two lists. The pairwise product is computed by multiplying every pair of elements from the original two lists, and storing each result in a new list.

For example, given the two lists `[3, 5, 2]` and `[2, 0, 2]`, the pairwise product is `[6, 0, 4]` which is equal to `[3 * 2, 5 * 0, 2 * 2]`.

You may assume that the lists passed to the function have the same length.

```
def pairwise_product(first, second):
    result = [None] * len(first)
    for i in __________(PART A)__________:
        ____________________(PART B)____________________
```

10. What is the correct code to insert in the blank marked `(PART A)`?

- ○ `len(first)`
- ○ `range(first)`
- ○ `range(len(first))`
- ○ `first`

11. What is the correct code to insert in the blank marked `(PART B)`?

- ○ `result[i] = first[i] * second[i]`
- ○ `result[i] = first * second`
- ○ `result = first[i] * second[i]`
- ○ `result = first * second`

12. Consider the following lines of code:

```
print('Bounce')
print('Bounce')
print('Bounce')
print('Bounce')
print('Bounce')
```

Write two lines of code that, when run, produce the same output.

13. Consider the following recursive function `func`. What is the output when the code below is run?

```
def func(a,b):
    if b == 1:
        return a
    return a * func(a,b-1)

print(func(2,4))
```

- ○ 8
- ○ 10
- ○ 16
- ○ 24

14. Li, Zaria, and Orr would like some juice. Ms. B uses the following function to serve them juice (by greeting them with a message):

```
def pour_juice():
        lst = ["Li", "Zaria", "Orr"]
        for name in lst:
                greeting = "Red juice for " + name + "!"
                print(greeting)

pour_juice()
# Output:
#     Red juice for Li!
#     Red juice for Zaria!
#     Red juice for Orr!
```

Now there are more TAs at JamCoders, and more juice colours! Ms. B is asking you to add arguments to her function so that she can greet any **list of TAs** with each day's **juice colour**. Fill in the blanks to add arguments and change the body of the function.

```
# YOUR CODE IN THE BLANKS
def pour_juice(________, ________):
    for name in ________:
        greeting = ________ + " juice for " + ________ + "!"
        print(greeting)
```

Your function should be able to generate the following outputs:

```
pour_juice(...)
# Output:
#     Green juice for Ian!
#     Green juice for Emaan!

pour_juice(...)
# Output:
#     Yellow juice for Micah!
#     Yellow juice for Pawat!
#     Yellow juice for Jonathan!
```

15. Consider the following lines of code:

```
def mystery(lst, item):
  if lst == []:
      return 0
  if lst[0] >= item:
      return 1 + mystery(lst[1:], item)
  else:
      return mystery(lst[1:], item)

print(mystery([2, 8, 8, 7, 10, 12, 6], 9))
```

What is the printed output of the code above?

- ○ `[10, 12]`
- ○ `22`
- ○ `[3, 9, 9, 8, 11, 13, 7]`
- ○ `2`

16. You are given a function `skip_add`. It takes in two integer arguments, n and k, and adds all numbers starting from n all the way to 0 jumping by k integers each time. For example:

```
Input:   skip_add(10, 3)
Returns:  22 # 10 + 7 + 4 + 1

Input:   skip_add(20, 6)
Returns:  44 # 20 + 14 + 8 + 2
```

Part of the `skip_add` function is written for you. Fill in the blank spaces to complete it. Draw a circle around the recursive case.

```
def skip_add(n, k):
   if ________________:        # YOUR CODE HERE
       return ____________     # YOUR CODE HERE
   return n + skip_add(n-k, k)
```

17. Consider the following code:

```
nums = [2, 3, 4, 5]
i = 0
while i < len(nums):
    print(nums[i])
    if nums[i] % 2 == 0:
        i += 3
    else:
        i -= 1
```

When it is run, what is printed?

- ○ 2 ↵ 3 ↵ 4 ↵ 5
- ○ 2 ↵ 5 ↵ 4
- ○ 2 ↵ 5 ↵ 4 ↵ 3
- ○ An error occurs.

18. The mathematician Mathixx has defined a new sequence, the "Mathixx Sequence," like so:

$$m_0 = 1$$
$$m_i = 3(m_{i-1}) - 1$$

where m_i is the i'th Mathixx Number. For example, the first few numbers in the sequence are as follows: $m_0 = 1, m_1 = 2, m_2 = 5, m_3 = 14$.

Write a recursive function `mathixx(i)` that takes an integer i and returns the i'th mathixx number, m_i. Your function should have the signature (where `i` is an `int`):

```
def mathixx(i):
    # YOUR CODE HERE
```

Write your function below.

19. The `triangle` function takes as input a string `s`. It prints `s`, then `s` with the first character removed, then `s` with the first two characters removed, and so on.

For example, `triangle("hello")` prints:

```
hello
ello
llo
lo
o
```

and `triangle("bye")` prints:

```
bye
ye
e
```

Your function should have the signature (where `s` is a `str`):

```
def triangle(s):
    # YOUR CODE HERE
```

Implement the `triangle` function below.

The following bonus problem is challenging. Please complete and **check** your answers to the previous problems before attempting to solve the bonus.

20. Bonus Problem:

You have 500 oranges, but one of them is fake. The fake one is heavier than the rest of them, while all the others are equal in weight.

You have access to a balancing scale that lets you weigh any group of oranges against any other group of oranges. The scale will tell you if one group of oranges is heavier than the other (and which group is heavier, if that is the case), or if both groups are of the same weight.

Your goal is to find the fake orange while minimizing the number of measurements you make with the scale. What procedure would you use? Describe your reasoning only in words. No code is needed.

According to your algorithm, how many times do you need to use the scale until you identify the fake orange?