



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 je():  
    return 'je'  
  
def ni():  
    print('la')  
    return 'ni'  
  
je()  
print(je())  
ni()  
print(ni())
```

What is the printed output of the code above?

- je ↵ la ↵ ni
- la ↵ ni
- je ↵ je ↵ la ni ↵ ni
- je ↵ la ↵ la ↵ ni

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 is printed after running the following lines of 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 ↵ 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)_____  
    return result
```

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. What is printed after running the following lines of code? Write your answers on the dashed lines.

```
def print_between(start, stop):
    i = start
    while i < stop:
        print(i)
        i += 1

print_between(5, 8) # Printed: -----
print_between(8, 5) # Printed: -----
```

14. James, 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 = ["James", "Zaria", "Orr"]
    for name in lst:
        greeting = "Red juice for " + name + "!"
        print(greeting)

pour_juice()
# Output:
#     Red juice for James!
#     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 Reggie!
#     Green juice for Ecy!

pour_juice(...)
# Output:
#     Yellow juice for Tarun!
#     Yellow juice for Xavier!
#     Yellow juice for Anita!
```

15. What is printed after running the following lines of code? Write your answers on the dashed lines.

```
def kim(x):
    print(x)
    return 'kim'

def ber(y):
    print('ber')
    return y

def li():
    return 'li'

kim('orr')           # Printed: -----
print(kim('orr'))    # Printed: -----
print(kim(ber(li()))) # Printed: -----
```

16. 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 `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 following signature (where `s` is a `str`):

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

Implement the `triangle` function below.

```
-----
-----
-----
-----
-----
-----
-----
-----
-----
```