Environmental Diagrams and Higher-Order Functions

Computer Science Mentors 61A

January 30–February 3, 2023

1 Environment Diagrams

1. When do we make a new frame in an environment diagram?

2. Give the environment diagram and console output that result from running the following code.

```
def swap(x, y):
    x, y = y, x
    return print("Swapped!", x, y)

x, y = 60, 1
a = swap(x, y)
swap(a, y)
```

Created by Alyssa Smith, Esther Shen, Maya Romero, Natalie Wei, Sandhya Ganesan, Kaelyn Huang, Cecilia Aiko, Aurelia Wang

3. Draw the environment diagram that results from running the following code.

```
def funny(joke):
    hoax = joke + 1
    return funny(hoax)

def sad(joke):
    hoax = joke - 1
    return hoax + hoax

funny, sad = sad, funny
result = funny(sad(2))
```

2 Higher-Order Functions

1. Why and where do we use lambda and higher-order functions? Can you give a practical example of where we would use a HOF?

2. Give the environment diagram and console output that result from running the following code.

```
x = 20
def foo(y):
    x = 5
    def bar():
        return lambda y: x - y
    return bar

y = foo(7)
z = y()
print(z(2))
```

3. Fill in the blanks (*without using any numbers in the first blank*) such that the entire expression evaluates to `9`.

```
(lambda x: lambda y: ____________________)(_____)(lambda z: z*z)()
```

4. Write a function, `whole_sum`, which takes in an integer, `n`. It returns another function which takes in an integer, and returns `True` if the digits of that integer sum to `n` and `False` otherwise.

```
def whole_sum(n):
    """
    >>> whole_sum(21)(777)
    True
    >>> whole_sum(142)(10010101010)
    False
    """
    def check(x):

        ____________________

        while _______________:

            last = ____________________

            ____________________________

            ____________________________

        return ____________________

    return ____________________
```

5. What would Python display?

```
def mystery(f, x):
    def helper(y):
        return f(x, y)
    return helper

>>> foo = mystery(lambda a, b: a(b), lambda c: 5 + square(c))
>>> foo(-2)
```

6. What would Python display?

   (a)
```
> (lambda x: x(x))(lambda y: 4)
```

   (b)
```
> (lambda x, y: y(x))(mul, lambda a: a(3, 5))
```