1 Functions

Questions

1.1 Determine what the Python interpreter will output given the following lines of code.

```python
>>> from operator import add, mul
>>> mul(add(5, 6), 8)
>>> print('x')
>>> y = print('x')
>>> print(y)
>>> print(add(4, 2), print('a'))
```

1.2 Determine what the Python interpreter will output given the following lines of code.

```python
>>> def foo(x):
    print(x)
    return x + 1

>>> def bar(y, x):
    print(x - y)

>>> foo(3)

>>> bar(3)

>>> bar(6, 1)

>>> bar(foo(10), 11)
```
2  Control

Questions

2.1 Which numbers will be printed after executing the following code?

```python
def count_digits(n):
    ...

    >>> count_digits(4)
    1
    >>> count_digits(12345678)
    8
    >>> count_digits(0)
    0
    ...
```
2.4 Define a function, count_matches, which takes in two integers n and m, and counts the number of digits that match.

```python
def count_matches(n, m):
    #
    >>> count_matches(10, 30)
    1
    >>> count_matches(12345, 23456)
    0
    >>> count_matches(121212, 123123)
    2
    >>> count_matches(111, 11)  # only one's place matches
    2
    >>> count_matches(101, 10)  # no place matches
    0
    #
```
3 Environment Diagrams

Questions

3.1 Draw the environment diagram for evaluating the following code

```python
def f(x):
    return y + x

y = 10
f(8)
```

3.2 Draw the environment diagram for evaluating the following code

```python
def dessef(a, b):
    c = a + b
    b = b + 1

b = 6
dessef(b, 4)
```
3.3 Draw the environment diagram for evaluating the following code

```python
def foo(x, y):
    foo = bar
    return foo(bar(x, x), y)

def bar(z, x):
    return z + y

y = 5
foo(1, 2)
```

3.4 Draw the environment diagram for evaluating the following code

```python
def spain(japan, iran):
    def world(cup, egypt):
        return japan-poland
    return iran(world(iran, poland))

def saudi(arabia):
    return japan + 3

japan, poland = 3, 7
spain(poland+1, saudi)
```
3.5 Draw the environment diagram for evaluating the following code

cap = 9
hulk = 3

def marvel(cap, thor, avengers):
    marvel = avengers
    iron = hulk + cap
    if thor > cap:
        def marvel(cap, thor, avengers):
            return iron
    else:
        iron = hulk
        return marvel(thor, cap, marvel)

def iron(man):
    hulk = cap - 1
    return hulk

marvel(cap, iron(3), marvel)