Guerrilla Section 1: Functions, Control, Environment Diagrams

Instructions

Form a group of 3-4. Start on Question 0. Check off with a staff member when everyone in your group understands how to solve the questions up to the first checkpoint. Repeat for the second checkpoint, the third checkpoint, and so on. **You're not allowed to move on after a checkpoint until you check off with a staff member.** You are allowed to use any and all resources at your disposal, including the interpreter, lecture notes and slides, discussion notes, and labs. You may consult the staff members, **but only after you have asked everyone else in your group. The purpose of this section is to have all the students working together to learn the material.**

Functions

Question 0:

What will Python output?

```
>>> from operator import add, mul
>>> mul(add(5, 6), 8)
```
88

```
>>> print('x')
```
x

```
>>> y = print('x')
```
x

```
>>> print(y)
```
None

```
>>> print(add(4, 2), print('a'))
```

a,

6 None

Question 1: Raising the Bar

What will Python output?

```
>>> 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)
```

3
4

Type error

-5

10
+1
0
→ 11 做为返回值

STOP!

Don't proceed until everyone in your group has finished and understands all exercises in this section, and you have gotten checked off!

Control

Question 2: Control yourself

a) Which numbers (1-4) will be printed after executing the following code?

```
n = 0
if n:
    print(1)
elif n < 2
    print(2)
else:
    print(3)
print(4)
```

2

4

b) WWPD (What would Python Display) after evaluating each of the following expressions?

```
>>> 0 and 1 / 0
```

error

```
>>> 6 or 1 or "a" or 1 / 0
```

6

```
>>> 6 and 1 and "a" and 1 / 0
```

error

```
>>> print(print(4) and 2)
```

4 none and 2

```
>>> not True and print("a")
```

false

Question 3: You have control

a) Define a function, `count_digits`, which takes in an integer, `n`, and counts the number of digits in that number.

```
def count_digits(n):
    """
    >>> count_digits(42)
    2
    >>> count_digits(12345678)
    8
    >>> count_digits(0)
    0
    """
    number = 0
    while n > 0:
        n = n // 10
        number += 1
    return number
```

b) Define a function, `count_matches`, which takes in two integers `n` and `m`, and counts the number of digits that match.

```
def count_matches(n, m):
    """
    >>> count_matches(10, 30)
    1
    >>> count_matches(12345, 23456)
    0
    >>> count_matches(212121, 321321)
    2
    >>> count_matches(101, 11) # only one's place matches
    1
    >>> count_matches(101, 10) # no place matches
    0
    """
    count = 0
    while (n and m) > 0 :
        if (n % 10 == m % 10):
            count += 1
        n = n // 10
        m = m // 10
    return count
```

STOP!

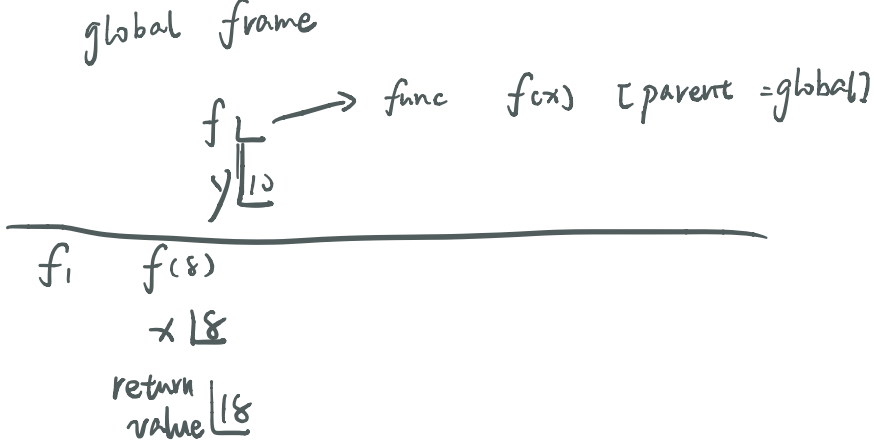
Don't proceed until everyone in your group has finished and understands all exercises in this section, and you have gotten checked off!

Environment Diagrams

Question 4: A New Environment

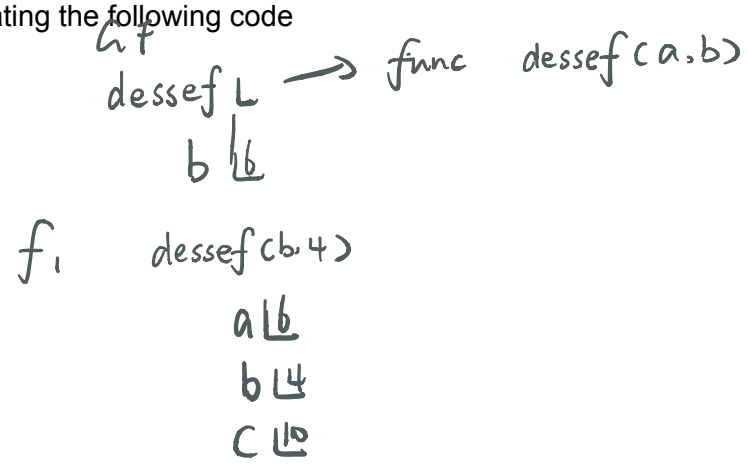
a) Draw the environment diagram for evaluating the following code

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

y = 10
f(8)
```



b) Draw the environment diagram for evaluating the following code

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

b = 6
dessef(b, 4)
```



STOP!

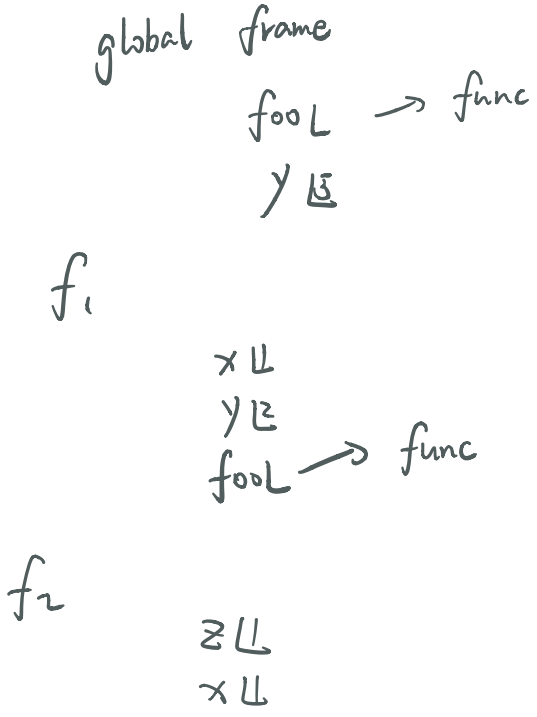
Don't proceed until everyone in your group has finished and understands all exercises

in this section, and you have gotten checked off!

Question 5: Environmental Collapse

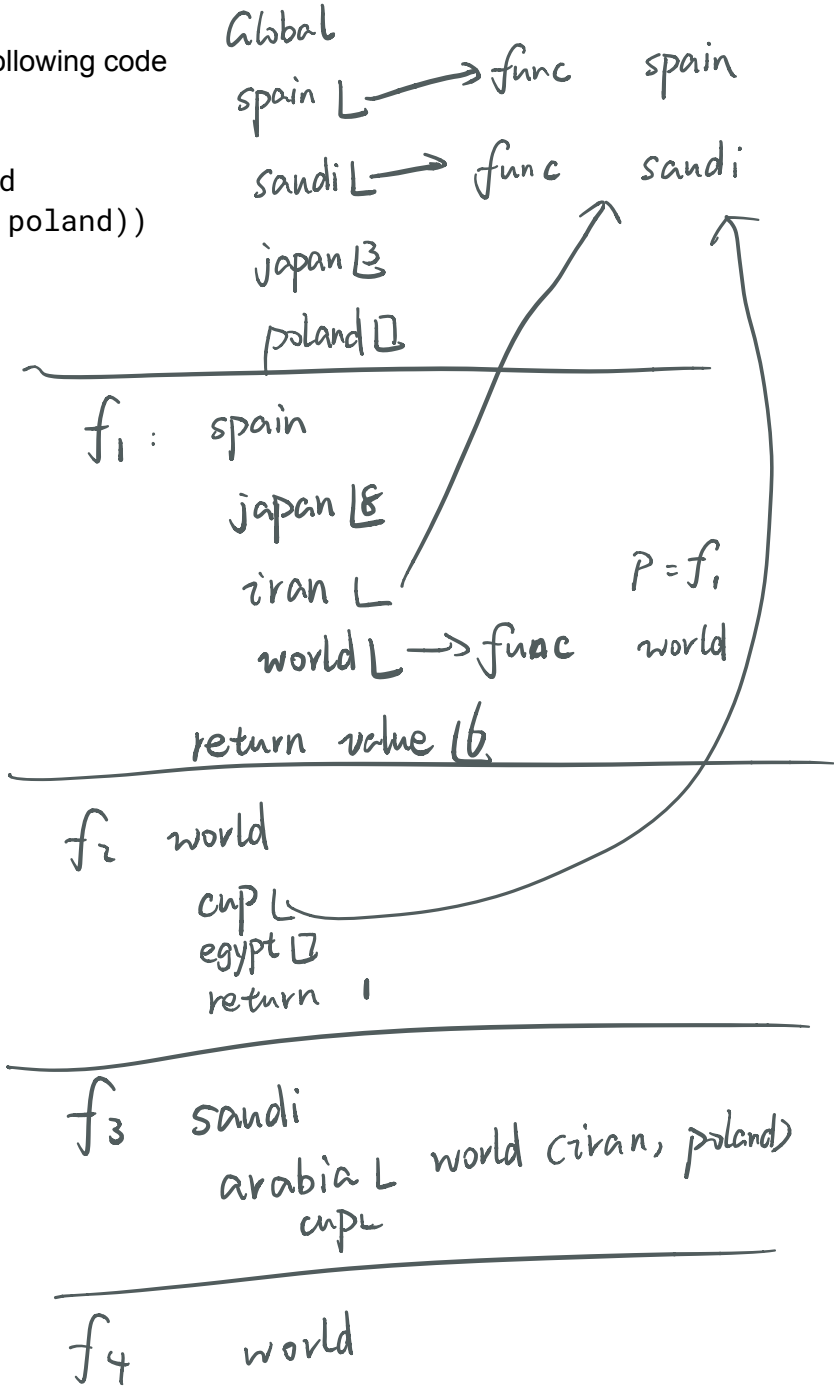
a) Draw an environment diagram for the following code

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

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

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



b) Draw an environment diagram for the following code

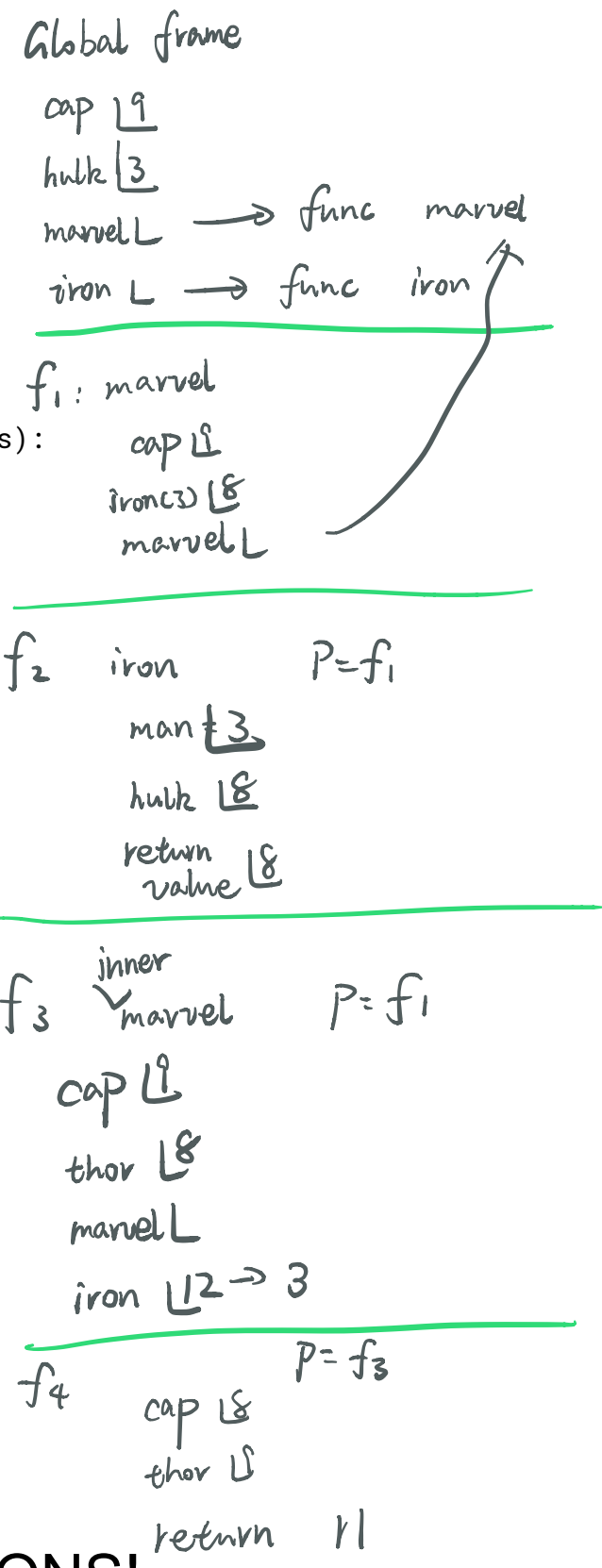
```
def spain(japan, iran):
      def world(cup, egypt):
            return japan-poland
      return iran(world(iran, poland))

def saudi(arabia):
      return japan + 3
japan, poland = 3, 7

spain(poland+1, saudi)
```



variable comes from where it was called

c) Draw an environment diagram for the following code

```
cap = 9
hulk = 3

def marvel(cap, thor, marvel):
      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)
```



CONGRATULATIONS!

You made it to the end of the worksheet! Great work :)