

Section #4: Midterm Review Solutions

Juliette Woodrow, Anna Mistele, John Dalloul, and Elyse Cornwall

Trace Problem

Num Operations

```
>>> 1 + 2 - 3 + 4
4
>>>
>>> 3 + 4 * 2
11
>>> 2 * (5 + 3)
16
```

Function Tracing

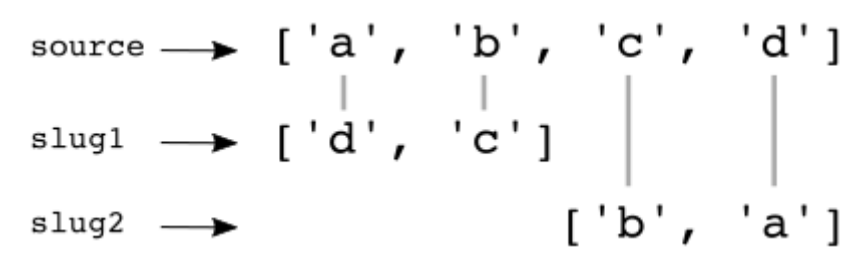
```
def b(y, x):
    z = x + y
    print("b variables: ", x, y, z)
    if z % 2 == 0: # if z is even
        return z + 1
    return z - 1

def main(): # code execution starts here
    x = 4
    y = 8
    z = x + y
    y = b(x, z)
    print("main variables: ", x, y, z)
```

b variables: 12 4 16
main variables: 4 17 12

Double Slug Encryption

Double-slug encryption uses one source list and two slug lists, all containing lowercase chars. The source list is even length, and the two slugs are each half its length. The slug1 list holds the encrypted form of chars in the first half of the source list. The slug2 list holds the encrypted form of chars in the second half of the source list. No char is in both slug1 and slug2. Here is an example with a length-4 source list:



Encrypt examples with source/slugs above:

The encrypt of 'a' is 'd'
The encrypt of 'c' is 'b'

```
def encrypt(source, slug1, slug2, ch):
    midway = len(source) // 2
    if ch in source:
        idx = source.index(ch)
        if idx < midway:
            return slug1[idx]
        else:
            return slug2[idx - midway]
    return ch
```

Grid Problem

```
def is_scared(grid, x, y):
    # check for person there
    if grid.get(x, y) == 'p' or grid.get(x, y) == 'y':
        # bear to the right!
        if grid.in_bounds(x + 1, y) and grid.get(x + 1, y) == 'b':
            return True
        # bear above!
        if grid.in_bounds(x, y - 1) and grid.get(x, y - 1) == 'b':
            return True
    return False

def run_away(grid):
    for y in range(grid.height):
        for x in range(grid.width):
            if is_scared(grid, x, y):
                grid.set(x, y, 'y') # begin yelling
                if grid.in_bounds(x - 1, y) and grid.get(x - 1, y) == None:
                    # move the yelling person one to the left
                    grid.set(x - 1, y, 'y')
                    grid.set(x, y, None)
```