Exercise 1

Bookmark this page

Exercise 1

11 points possible (graded)

ESTIMATED TIME TO COMPLETE: 12 minutes

In this problem, we'll examine how indirection works. Consider the following definitions:

```
a = [1, 2, 3, 4, 0]
b = [3, 0, 2, 4, 1]
c = [3, 2, 4, 1, 5]
```

1. What is the value of the following expressions? If you think there will be an error, please type in 'error' (without quotes) in the input box.

    1. `a[0]`

    2. `b[1]`

    3. `a[a[1]]`

    4. `b[b[2]]`

5. `a[b[2]]`

6. `c[a[b[3]]]`

7. `a[c[a[b[0]]]]`

8. `a[c[a[b[3]]]]`

2. Assume we have defined the following function:

```
def foo(L):
    val = L[0]
    while (True):
        val = L[val]
```

Which of the following statement(s) will result in an infinite loop?

- [ ] `foo(a)`
- [ ] `foo(b)`
- [ ] `foo(c)`

3. Consider the following code:

```
num = ???
L = [5, 0, 2, 4, 6, 3, 1]
val = 0
for i in range(0, num):
    val = L[L[val]]

print(val)
```

1. What is the smallest value that `num` can be such that the number 3 is printed?

- 0
- 1
- 3
- 5
- Impossible

2. Now, we redefine `L` to be:

```
L = [2, 0, 1, 5, 3, 4]
```

What is the smallest value that `num` can be such that the number 3 is printed?

- 0
- 3
- 5

Impossible

Submit

Exercise 1
Topic: Lecture 12 / Exercise 1

Show Discussion