Fall 2022

LDOC - Recursion Practice

Announcements

- EX11 Due Tomorrow, 11/30 at 11:59pm
  - Effectively optional with one exercise drop added to syllabus.
  - Students continuing to COMP210 Data Structures will find this exercise valuable!

- Office Hours through Tomorrow at 5pm

- Final Exam - In-person Friday at 12pm
  - Section 001 - Hamilton 100
  - Section 002 - Split between Chapman 211 and Carroll 111
  - Seat assignments will go on Sakai's Postem Tool by Thursday (and I will post announcement)
  - Last day to submit exam conflict form! (Google: UNC Exam Excuse Request Form.)
  - Make-up Final Exam for Approved Alternates - 8am Sunday 12/4 in SN011

Code Writing Practice

- Write a class with the following characteristics:

- The class' name is Staff.
- Every Staff object has two attributes: name (string) and is_cs (bool).
- You should be able to construct a Staff object with a constructor that has parameters to initialize each attribute
- You should implement any methods necessary to implement the following behavior:

```
>>> prof: Staff = Staff("Kris", True)
>>> print(prof.greet())
Hello, I'm Kris in CS
>>> dr: Staff = Staff("Mara", False)
>>> print(dr.greet())
Hello, I'm Mara NOT in CS
```

Write a class with the following characteristics:

The class' name is **Staff**.

Every **Staff** object has two attributes: **name** (string), and **is_cs** (bool).

You should be able to construct a Staff object with a constructor that has parameters to initialize each attribute

You should implement any methods necessary to implement the following behavior:

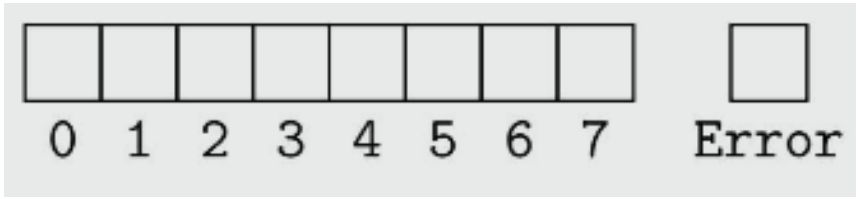
```
>>> prof: Staff = Staff("Kris", True)
>>> print(prof.greet())
Hello, I'm Kris in CS
>>> dr: Staff = Staff("Mara", False)
>>> print(dr.greet())
Hello, I'm Mara NOT in CS
```

In this series of questions, you will trace code that modifies a boolean `list a`. You will respond beneath each code listing by *completely shading in the squares of items whose value is assigned* ***`True`***. If an error occurs during the evaluation of the loop, fill in the `Error` box and stop evaluating. If any item's value was assigned `True` prior to the error, keep its value shaded in.

You can assume `a` is initialized with ***`8 False`*** *elements*, as shown below, and that each question is independent of the next.

```
f: bool = False
a: list[bool] = [f, f, f, f, f, f, f, f]
```

```
4    i: int = 0
5    while i < len(a):
6        if i % 2 == 1 and i >= 3:
7            a[i] = True
8        i += 1
```




```
4    i: int = 0
5    while i <= 8:
6        if i % 2 == 0:
7            a[i] = True
8        i += 1
```

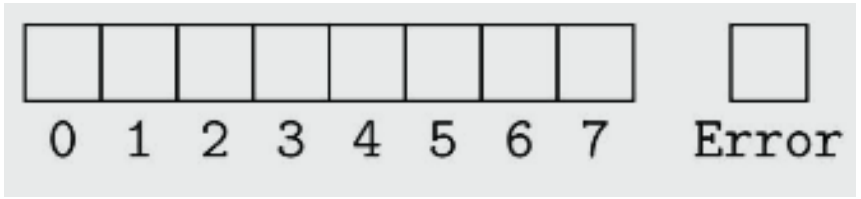


Diagram the code listing to the right.

Be sure to leave a lot of room for output!

```
1   def branch(a: float, l: float) -> None:
2       print("branch!")
3       print(f"a {a} - l {l}")
4
5       if l < 4.0:
6           ...  # DO NOTHING!
7       else:
8           branch(a + 45.0, l * 0.5)
9           branch(a - 45.0, l * 0.25)
10
11      print(f"a {a + 180.0} - l {l}")
12
13
14  angle: float = 90.0
15  length: float = 4.0
16  branch(angle, length)
```

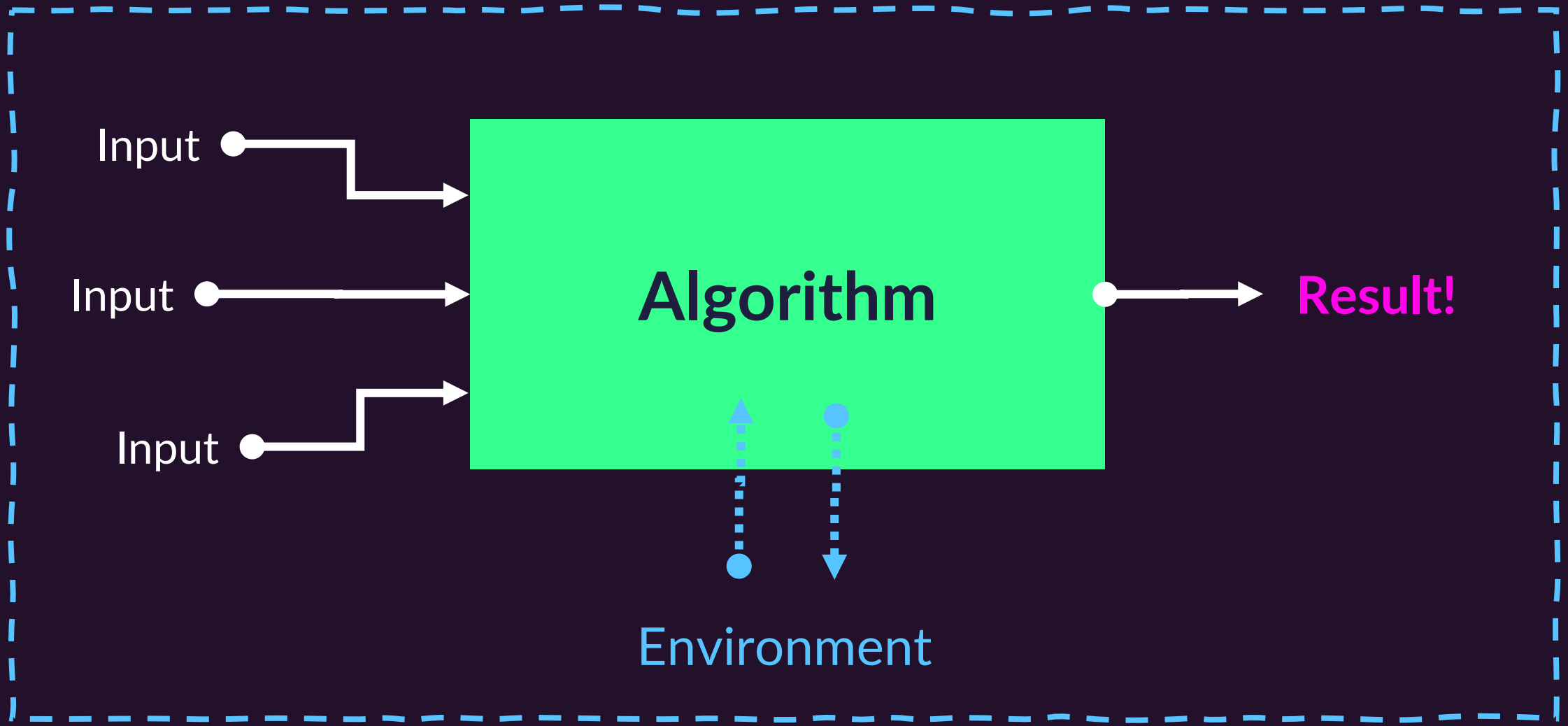
```python
def branch(a: float, l: float) -> None:
    print("branch!")
    print(f"a {a} - l {l}")

    if l < 4.0:
        ...  # DO NOTHING!
    else:
        branch(a + 45.0, l * 0.5)
        branch(a - 45.0, l * 0.25)

    print(f"a {a + 180.0} - l {l}")


angle: float = 90.0
length: float = 4.0
branch(angle, length)
```

The Fundamental Pattern



Thank **YOU** *for a great semester!*

COMP 110