CL10 – Reference Types, Practice with Lists, and Nested while Loops

Announcements

- EX02 – Wordle due Sunday, June 1 at 11:59pm
- Please review your graded Quiz 01 as prep for Quiz 02!
- CQ03 – Memory Diagram – due today at 11:59pm
- Quiz 02 tomorrow! Ways to prep:
  - Review anything you missed on Quiz 01
  - Do practice quiz 02
  - Ask us for help in Office Hours:
    - In-person: 1-3pm today
    - Virtual: 3-6pm today
  - **Review Session via the virtual Office Hours Zoom link at 6pm tonight!**

Warmup: In VS Code, write the following code:

1. Create an *explicitly typed* list called **`names`**, with the initial values: “Rosie”, “Abu”, and “Sally”
2. Add “Jade” to the end of the list using a method call
3. Remove “Sally” from the list using a method call
4. Print out the number of items in the list
5. Use subscription notation and the list to print the name “Abu”
6. If the name “Sally” is in the list, print “found Sally!”, otherwise, print “Not found”

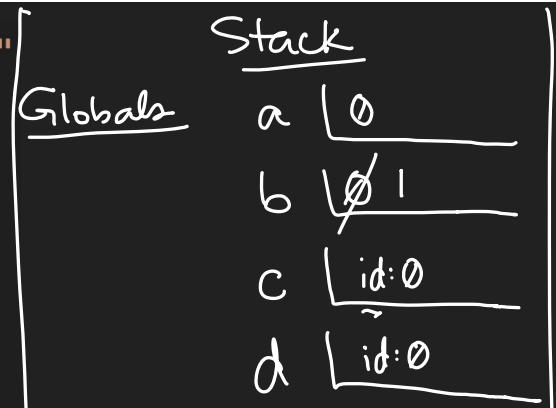
```python
"""An example of primitive vs. reference types."""

a: int = 0
b: int = a
b += 1
print(f"a is: {a}")
print(f"b is: {b}")

c: list[int] = [b, 4]
d: list[int] = []
c = d
d.append(13)

print(c)
```

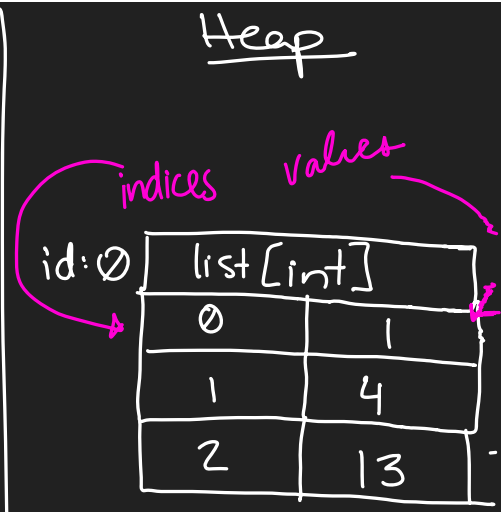
Output

a is: 0

b is: 1

[1, 4, 13]





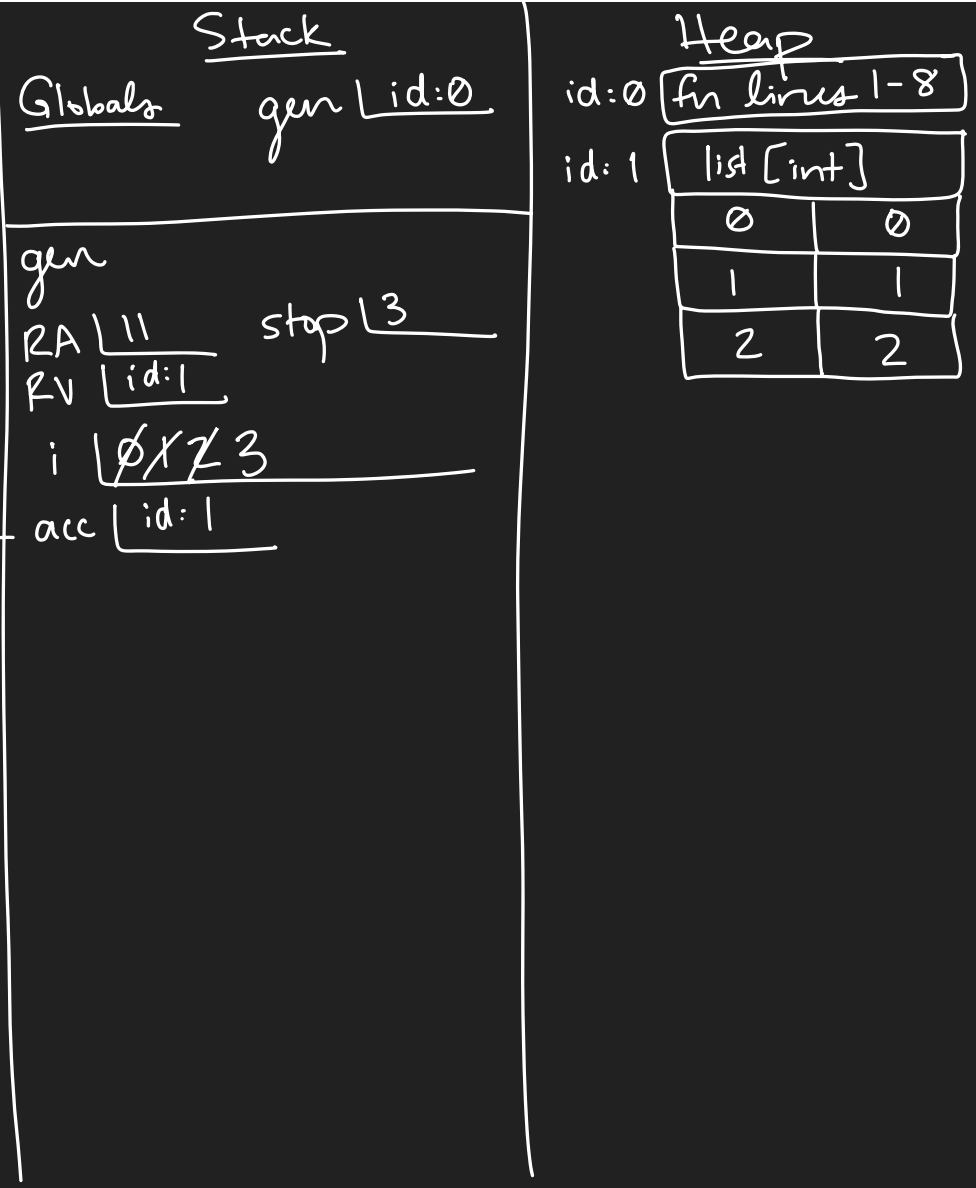
Your turn: write a function called **gen** that takes as input an int called **n**, and returns a list of integers from 0 up to, but not including **n**

```
def gen(stop: int) -> list[int]:
    """Generate a list from 0 to stop, exclusive."""
    i: int = 0
    acc: list[int] = []
    while i < stop:
        acc.append(i)
        i = i + 1

    return acc
```

```python
def gen(stop: int) -> list[int]:
    """Generate a list from 0 to stop, not inclusive."""
    i: int = 0
    acc: list[int] = []
    while i < stop:
        acc.append(i)
        i = i + 1
    return acc


print(gen(3))
```

Output

[0, 1, 2]



Diagramming a Nested while Loop

```
1   def triangle(n: int) -> None:
2       i: int = 1
3       line: str
4       while i <= n:
5           line = ""
6           while len(line) < i:
7               line += "*"
8           print(line)
9           i += 1
10
11
12  triangle(2)
```

Output

Stack

Globals

triangle | id:0

triangle

RA | 12

RV | None

n | 2

i | 1 2 3

line | "" "*" "" "*" "**"

Heap

id:0 fn lines 1-9

Diagramming a Nested List (Submit this diagram for CQ03!)

```python
def sum2d(xs: list[list[int]]) -> int:
    """Calculate the sum of a 2-dimensional list of lists."""
    total: int = 0
    row_i: int = 0
    while row_i < len(xs):
        col_i: int = 0
        while col_i < len(xs[row_i]):
            total += xs[row_i][col_i]
            col_i += 1
        row_i += 1
    return total


values: list[list[int]] = [[1, 2, 3], [3, 4, 5]]
print(sum2d(values))
```

Output

18