Tutorial: Write a function similar to keep_ints like before, but now it takes in a number n and returns a function that has one parameter cond. The returned function prints out numbers from 1 to n where calling cond on that number returns True.

① prints ② if ③ while ④ counter

```
def make_keeper(n):
    """Returns a function which takes one parameter cond and prints out
    all integers 1..i..n where calling cond(i) returns True.

    >>> def is_even(x):
    ...     # Even numbers have remainder 0 when divided by 2.
    ...     return x % 2 == 0
    >>> make_keeper(5)(is_even)
    2
    4
    """
```

```
def con(cond):                          ✓
    counter = 1                         2 functions:
    while (counter <= n)                    1 inside another
        if (cond(counter))
            print counter
        counter += 1
    return con                          ✓
```

010

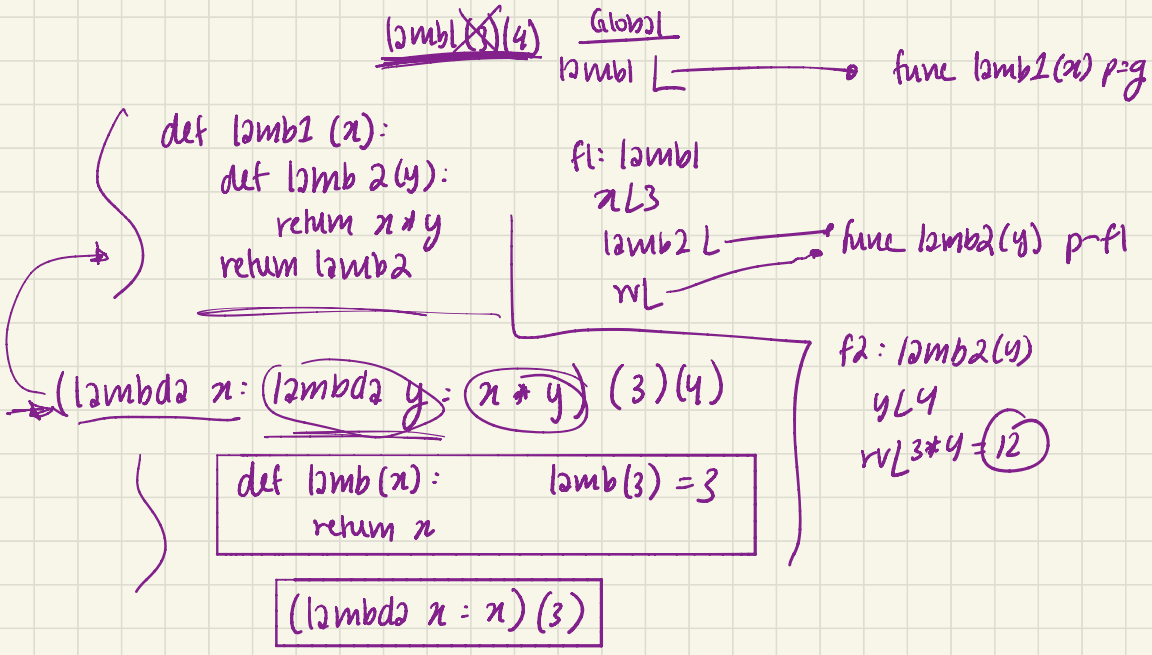
flip with bit

0 << 2

000

1.5 **Tutorial:** Draw the environment diagram that results from executing the code below.

```
1  n = 7
2
3  def f(x):
4      n = 8
5      return x + 1
6
7  def g(x):
8      n = 9
9      def h():
10         return x + 1
11     return h
12
13 def f(f, x):
14     return f(x + n)
15
16 f = f(g, n)
17 g = (lambda y: y())(f)
```



Question 6

```
def albert(albert):
    albert = albert()
    def albert():
        albert = lambda albert: albert
        return albert(albert)
    return albert

albert(lambda: albert)()
```

Toggle Solution