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. 3 while (i) counter (1) prinks 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): 2 functions: 1 justed 2 nother counter = 1 -1 while (counter < = n) 2 if (cond (counter)) Ч print country 5 counter t= (6 return con 010 (We with

```
1.5 Tutorial: Draw the environment diagram that results from executing the
           code below.
        1 n = 7
         2
           def f(x):
              n = 8
              return x + 1
           def g(x):
              n = 9
         8
         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)
                             (2mbl (3)(4)
                                             Global
                                                                 · func 12mb 2 (21) p=g
                                          12mbl 1-
           def lamb1 (n):
                                           fl: lambl
                det lamb 2(y):
                                             nL3
                    return nyy
                                                               func lambally) p-fl
                                              12mb2L
                return lamba
                                               W _-
                                                                f2: 12mb2(4)
                                              (3)(4)
(lambda n: (lambda y: n + y)
                                                                  464
123*4 = 12
```

lamb (3) = 3

def lamb (n):

return n

(lambda n = n) (3)

```
Question 6
    def albert(albert):
       albert = albert()
      def albert():
          albert = lambda albert: albert
          return albert(albert)
       return albert
1 albert(lambda: albert)()
   Toggle Solution
  From Frame
                              func albert (albert) p=g
     albert L-
    n1.
  fl: albut (albort)
     albut 1 -
 th: 1() p=9
f3: albert() p>f1
                           21hor (21box) p=f3 a
     albert L-
                 p= +3
fu: 2(21box)
   albutt.
    W
```