Homework 3
Due: Wednesday, March 23rd, 2022 11:59 PM EST

Tomas Hornicek, no collaborators,no sources,no extension

1 Without Coding [Show your work when asked]

Problem 1 (5 points each)

Calculate the result of the following function calls. Make sure to **show your work** tracing the re cursive calls.

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
```

[*1*]

```
def x(n):
    if n <= 1:
        return 1
    else:
        return n*(n-1)*x(n-1)
def y(m):
    if m == 1:
        return 1
    else:
        n = x(m - 1) #x(7) calls defx(n)
        return n
```

[*2*]

```
# Function that finds the nth tribonacci number
def tribonacci(n):
```

```
if n == 1:
    return 0
elif n == 2:
```

[*3*]

```
    return 1
elif n == 3:
    return 2
else:
    return tribonacci(n-1) + tribonacci(n-2) + tribonacci(n-3)
```


Problems to Trace

1. factorial(5)

Answer: ->factorial(5) -> 5*factorial(4) ->4*factorial(3) ->3*factorial(2) ->2*factorial(1) ->1*factorial(0) ->factorial(0) returns 1 -> factorial(1) returns 1*1 -> factorial(2) returns 2*1 ->factorial(3) returns 3*2 ->factorial(4) returns 4*6 -> factorial(5) returns 5*24 ->120

2. factorial(8)

Answer: factorial(8)->8*factorial(7)->7*factorial(6)->6*factorial(5) -> 5*factorial(4) ->4*factorial(3) ->3*factorial(2) ->2*factorial(1) ->1*factorial(0) ->factorial(0) returns 1 -> factorial(1) returns 1*1 -> factorial(2) returns 2*1 ->factorial(3) returns 3*2 ->factorial(4) returns 4*6 -> factorial(5) returns 5*24 ->factorial(6) returns 6*120-> factorial(7)returns 7*720 ->factorial(8) returns 8*5040 ->40 320

3. y(8)

Answer: y(8)->n=x(8-1)->x(7)->7*6*x(6)-> 6*5*x(5)-> 5*4*x(4)-> 4*3*x(3)-> 3*2*x(2)-> 2*1*x(1)->2-> 3*2*2 -> 4*3*12-> 5*4*144-> 6*5*2880-> 7*6*86400-> 3 628 800

4. tribonacci(4)

Answer: tribonacci(4)-> tribonacci(4-1) + tribonacci(4-2) + tribonacci(4-3)-> 2+1+0 -> 3

5. tribonacci(7)

Answer: tribonacci(7)-> tribonacci(7-1) + tribonacci(7-2) + tribonacci(7-3) -> tribonacci(6) + tribonacci(5) + tribonacci(4)

tribonacci(4)-> tribonacci(4-1) + tribonacci(4-2) + tribonacci(4-3)-> 2+1+0 -> 3

tribonacci(5)-> tribonacci(5-1) + tribonacci(5-2) + tribonacci(5-3)-> tribonacci(4) + tribonacci(3) + tribonacci(2) -> 3+2+1 -> 6

tribonacci(6)-> tribonacci(6-1) + tribonacci(6-2) + tribonacci(6-3)-> tribonacci(5) + tribonacci(4) + tribonacci(3) -> 6+3+2-> 11

tribonacci(6) + tribonacci(5) + tribonacci(4) -> 11 + 6 + 3 -> 20

Problem 2 (10 points)

Find the expected result of the two list comprehension by hand (**show your work**):

list_1 = [10 - thing for thing in range(20,1,-1)]

Answer: The for loop iterates starting at 20, in steps of -1, ending iteration at 2. Each iteration will add 10 - the current iterated value to the list. Therefore the result will be: list_1 = [-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8]

list_2 = [val**3 for val in [1,1,2,2,3,3,4,4] if val!=2] [4]

Answer: The for loop iterates over the values in the inner list, excluding the values of 2. With each iteration the value in the list is raised to the power of 3 and added to list_2. Therefore the results will be: list_2 = [1,1,27,27,64,64]