CS 88 Spring 2020 Computational Structures in Data Science

Final Exam

INSTRUCTIONS

This is your exam. Complete it either at exam.cs61a.org or, if that doesn't work, by emailing course staff with your solutions before the exam deadline.

This exam is intended for the student with email address `zacksoule@berkeley.edu`. If this is not your email address, notify course staff immediately, as each exam is different. Do not distribute this exam PDF even after the exam ends, as some students may be taking the exam in a different time zone.

For questions with **circular bubbles**, you should select exactly *one* choice.

- ◯ You must choose either this option
- ◯ Or this one, but not both!

For questions with **square checkboxes**, you may select *multiple* choices.

- ☐ You could select this choice.
- ☐ You could select this one too!

You may start your exam now. Your exam is due at 11:00AM Pacific Time. Go to the next page to begin.

Instructions

0.1 CS88 Spring 2020 Final Exam

0.1.1 *Section 1*

Please review all instructions before begining the exam.

If you have questions, please post a private note on Piazza. Pleasr do not use emaul.

This is exam is *strictly closed to collaboration.* All work must be your own! You make use of internet resources, course notes, Python Tutor, and a local Python interpreter.

(a) As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others.

UC Berkeley Honor Code

By signing my name below, I am agreeing to abide by the UC Berkeley Honor Code.

I agree that all the work turned in today is my own work and was completed according to the rules above.

(b) What is your student ID number?

1. (15 points) What Would Python Display?

0.1.2 Live Clarifications Document

For each of the expressions below, write a valid Python expression that would generate the output shown. The output may have multiple lines. The first two rows have been provided as examples.

The interactive interpreter displays the value of a successfully evaluated expression, unless it is `None`.

Assume that you have first started `python3` and executed the statements below.

There are many possible correct answers, but each answer should follow the restrictions provided and fit within the provided blanks, as well as be syntatically valid. Partial credit will be avaiable.

Note that the *length* of the blank input is meaningless, only that some expression should go there.

```
def f1(x, y):
    print(x + y)
    return y + x

lst1 = list(range(5))

class Act:
    def __init__(self, x):
        self.y = x

    def update(self, y):
        self.x = y
        return self.y
    def swap(self, y):
        self.x, self.y = self.x + y, self.y - y
    def __next__(self):
        self.x += self.y
        if self.x > 10:
            raise StopIteration
        return self.x
    def __iter__(self):
        self.x - 1
        return self
```

Question	Code	Python Output
Example	`f1(___)` Possible Answer: `f1(-1, 1)`	`0 0`
1.1	`f1(_______)`	`4 4`
1.2	`f1(_______)`	`[3, [1]] [[1], 3]`
1.3	`lst1[_______] + _______`	`[2, 3] [1, 2, [3, 2]]`
1.4	`act = Act(1)Act.swap(act, _______) [act.x, act.y]`	`[7, 0]`
1.5	`it = Act(2)print(it.update(3)) it.swap(-1) [_______ for x in it]`	`2 [11, 17]`

(a) **(3 pt)** Include the entire line: `f1(_______)`.

Note that in the PDF form of the exam this is not correctly rendered as 2 rows.

(b) **(3 pt)** Include the entire line: `f1(_______)`.

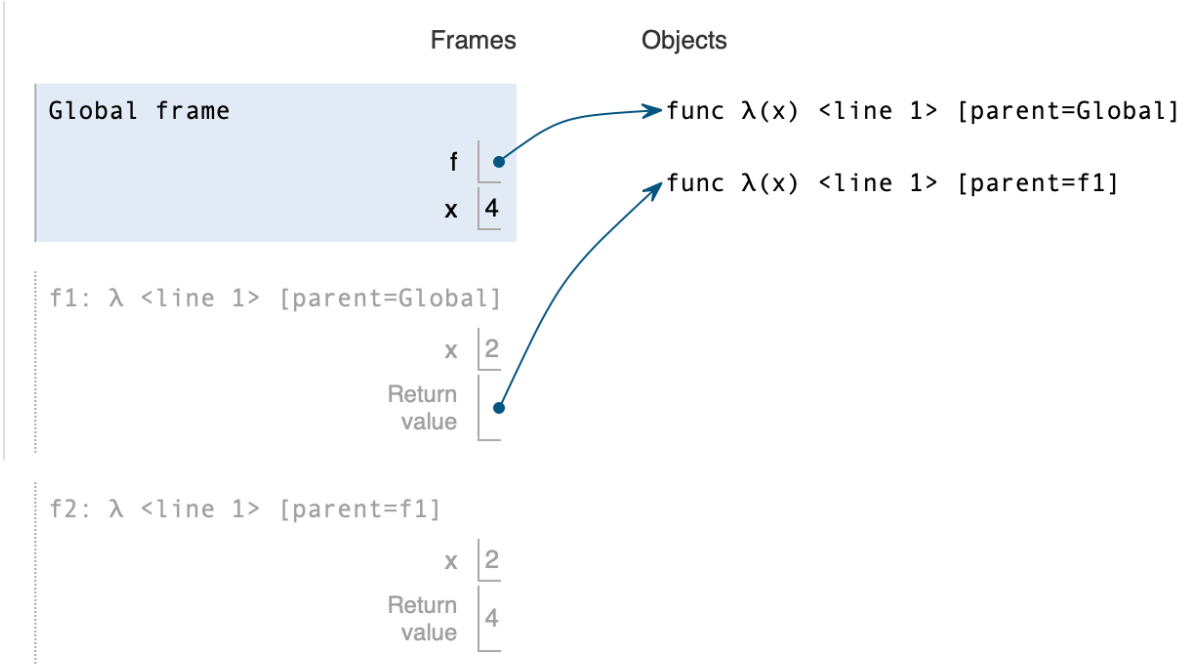
(c) **(3 pt)** Inlcude the entire line `lst1[_______] + _______`.

(d) **(3 pt)** Include the entire line `Act.swap(act, _______)`.

(e) **(3 pt)** Include the entire line `[_______ for x in it]`.

2. (8 points) An Environmental Distater!

Given the following environment diagram frames, please fill in the blanks so that your code will look like the provided environment diagram. There are many possible answers for each part.



Environment Diagram Output.

The correct solution will follow the form below. No additional lines are necessary.

```
f = ________
x = ________
```

Python Tutor Link

(a) (4 pt) `f = _____________`

(b) (4 pt) `x = _____________`

(c) (0 pt) Please provide a link to your environment diagram on tutor.cs61a.org. (The link will say pythontutor.org)

3. (12 points) Squash the Bugs!

Each of the functions below have a specific number of bugs. Read through the functions and doctests. First, write down what the function currently returns with the input given. After that, write down each bug you find and how to fix it. Please do not write down an alternate solution, rather, you should try to find bugs in the solutions provided. It may be helpful to include the line numbers in the description of your fixes.

What the code currently returns may or may not be the same as what is shown on the doctests.

Part 1: How often?

```
def frequency(text):
    """
    Returns a dictionary containing how often each word in the text appears.
    Do not worry about punctuation/ multiple whitespaces. Assume split() splits the text string into a
    Hint: one of the bugs would require adding a line of code with a function call.
    >>> text = ''bananas and apples and ApPlEs and oranges''
    >>> frequency(text)
    {''bananas'': 1, ''and'':3, ''apples'':2, ''oranges'':1}
    >>> text = ''cs cs 88''
    >>> frequency(text)
    {''cs'': 2, ''88'':1}
    """
    dict_word = {}
    textList = text.split()
    for word in textList:
        if word in dict:
            dict_word[word] = dict_word[word] + 1
            return dict_word
        dict_word[word] = 1
    return dict
```

(a) (2 pt) What does the code currently return with no changes?

```
frequency(''cs cs 88'')
```

(b) (4 pt)

Demonstrarte the bug. In the space provided, create a *new* input we haven't shown here.

You answer should contain the following: * An example call to `frequency` * The answer currently returned by the code above * What the code should actually return.

(c) **Fix the Bugs.**

For each part list one of the bugs present in the function above. The order of the 3 bugs does not matter.

i. (2 pt) 1 of 3. What is this bug and how would you fix it? Describe what code you would change.

ii. (2 pt) 2 of 3. What is this bug and how would you fix it? Describe what code you would change.

iii. (2 pt) 3 of 3. What is this bug and how would you fix it? Describe what code you would change.

4. (6 points) Just Bugs...Not Murder Hornets

Find the bugs in the following code. Use the examples as a guide for what *should* happen.

```
class Exam():
    course = 'cs88'
    instructors = ['Gerald Friedland', 'Michael Ball']
    tas = ['Alex Kassil', 'Brian Mi', 'Julia Yu', 'Alec Kan', 'Cameron Malloy', 'Shreya Kannan', 'Soph
    time = 90
    def __init__(self, name, student_id):
        self.name = name
        self.sid = student_id
        self.score = {}

    def grade(self, points_correct, points_total):
        grade = points_correct/points_total
        self.score[student_id] = grade

    def view_score(self):
        return score

class Question():
    extra_credit = False
    def grade(self, points_correct, points_total):
        if self.extra_credit == False:
            super().grade(self, points_correct, points_total)
        else:
            self.score[self.sid] += points_correct
```

```
>>> oski_exam = Exam('Oski Bear', 123456789)
>>> oski_q1 = Question('Oski Bear', 123456789)
>>> oski_q1.grade(24,36)
>>> oski_q1.score
{123456789: 0.6666666666666666}
>>> oski_exam.score
{}
```

For each part list one of the bugs present in the function above. The order of the 3 bugs does not matter.

(a) (2 pt) 1 of 3. What is this bug and how would you fix it? Describe what code you would change.

(b) (2 pt) 2 of 3. What is this bug and how would you fix it? Describe what code you would change.

(c) (2 pt) 3 of 3. What is this bug and how would you fix it? Describe what code you would change.

5. (9 points) Summoner's School for SQL

As a student of CS88, you've been noticed for your sharp SQL skills and magical powers in coding by the legends of Summoner's Rift. They offer you the title of "Master of Data," a title renowned by many, but only if you can complete the following SQL challenges. To aid you in your mission, you have given two tables: champions and positions.

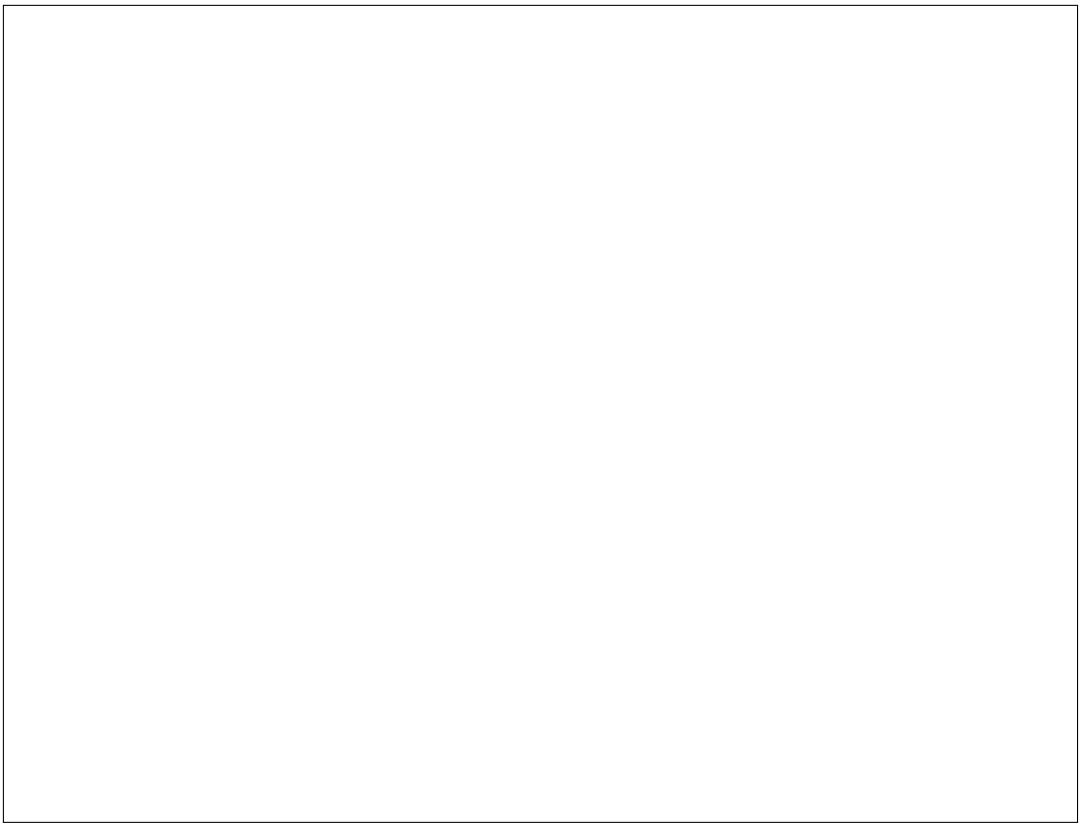
```
CREATE TABLE champions AS
    SELECT "Aatrox" AS name, 580 AS health, 38 AS armor UNION
    SELECT "Ahri", 526, 20.88 UNION
    SELECT "Akali", 575, 23 UNION
    SELECT "Alistar", 575, 44 UNION
    SELECT "Amumu", 613.12, 33 UNION
    SELECT "Ashe", 539, 26 UNION
    SELECT "Jax", 592.8, 36 UNION
    SELECT "Lux", 490, 18.72 UNION
    SELECT "Malphite", 574.2, 37 UNION
    SELECT "Vayne", 515, 23 UNION
    SELECT "Yasuo", 523, 30;

CREATE TABLE positions AS
    SELECT "Aatrox" AS name, "Top" AS position UNION
    SELECT "Ahri", "Middle" UNION
    SELECT "Akali", "Top" UNION
    SELECT "Akali", "Middle" UNION
    SELECT "Alistar", "Support" UNION
    SELECT "Amumu", "Jungle" UNION
    SELECT "Ashe", "Bottom" UNION
    SELECT "Jax", "Top" UNION
    SELECT "Jax", "Jungle" UNION
    SELECT "Lux", "Middle" UNION
    SELECT "Lux", "Support" UNION
    SELECT "Malphite", "Top" UNION
    SELECT "Malphite", "Jungle" UNION
    SELECT "Malphite", "Support" UNION
    SELECT "Vayne", "Top" UNION
    SELECT "Vayne", "Bottom" UNION
    SELECT "Yasuo", "Top" UNION
    SELECT "Yasuo", "Mid";
```

(a) (3 pt) Write a query that selects the name and health of the champion with the greatest health. You may assume health values are unique.

```
sqlite> CREATE TABLE health AS

    ------------------------------------------------------

    ------------------------------------------------------;

sqlite> SELECT * FROM health;
Amumu | 613.12
```

(b) (3 pt) Write a table which is the output of the following query. (As long as we can tell which are rows and columns, the specific syntax does not matter.)

```
CREATE TABLE roles AS
    SELECT position, COUNT(*) as count FROM positions
    GROUP BY position ORDER BY count DESC;
```



(c) **(3 pt)** Write a query to find all the unique pairs of champions who play in bottom position and champions who play in support position.

```
CREATE TABLE pairs AS

    ------------------------------------------------------

    ------------------------------------------------------;
```

```
sqlite> SELECT * FROM pairs;
Ashe | Alistar
Ashe | Lux
Ashe | Malphite
Vayne | Alistar
Vayne | Lux
Vayne | Malphite
```

6. (12 points) Iteration Generates Success

Given a list of single digits, create an Iterator that iterates through all the possible two digit numbers that can be created from digits in the list. The `__init__` method and `__iter__` method have been completed for you.

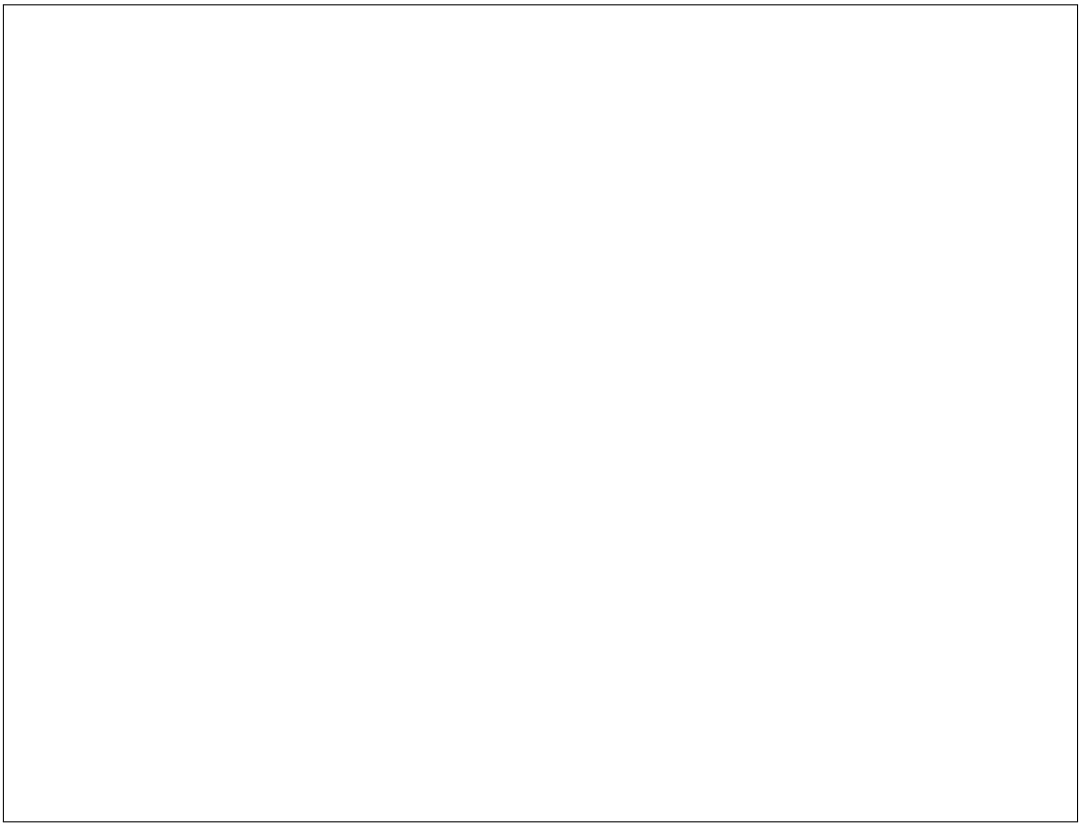
```
class BuildNum:
    """
    >>> for i in BuildNum([1, 2, 3]):
    ...     print(i)
    ...
    11
    12
    13
    21
    22
    23
    31
    32
    33
    """
    def __init__(self, lst):
        self.lst = lst
        self.i = 0
        self.j = 0

    def __next__(self):
        """YOUR CODE HERE"""

    def __iter__(self):
        return self
```

(a) (8 pt) Finish the body of the `__next__` method, such that the iterator works correctly.'

Use the following lines of code as the bases for your solution. The following lines of code are provided out of order. You may need to reorder the lines and fill in the blanks. A reasonable solution exists which uses all line, but there may be a solution which does not use all lines. There may be multiple correct solutions. The length of blank inputs has *no meaning on what expressions might fill in those blanks.*

```
def __next__(self):
   """YOUR CODE HERE"""
```

Lines of Code
`return result`
`if __________ == ______________:`
`if _________________ == len(self.lst):`
`self.i __________`
`self.j __________`
`else:`
`else:`
`result = _____________ * 10 + ____________`
`self.j += 1`
`raise StopIteration`

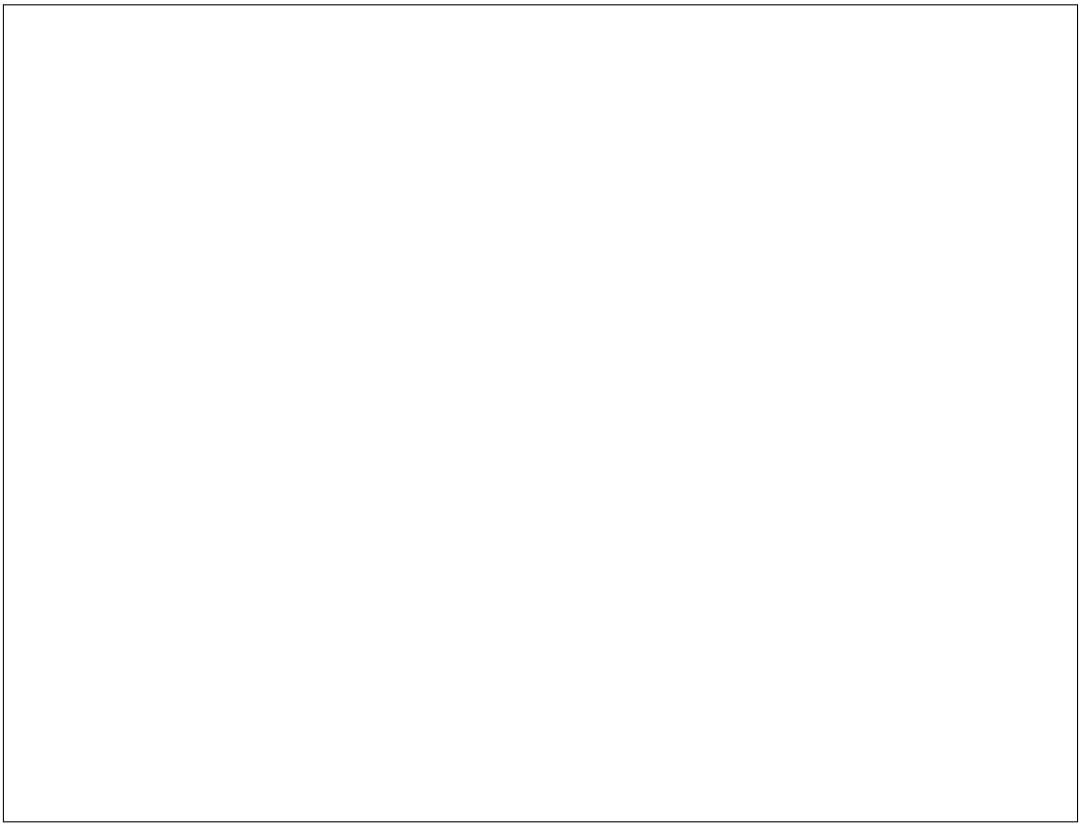


(b) **(4 pt)** Given a list of single digits, create a generator function that yields all the two digit numbers that can be created from the given list of digits and only have repeated digits, or a repdigit (ie. 11, 22, 33, ... 99). You may use BuildNum and assume it has been implemented correctly for this part.

Use the following lines of code as the bases for your solution. The following lines of code are provided out of order. You may need to reorder the lines and fill in the blanks. A reasonable solution exists which uses all line, but there may be a solution which does not use all lines. There may be multiple correct solutions. The length of blank inputs has *no meaning on what expressions might fill in those blanks.*

```
def rep_digit(lst):
    """
    >>> for i in rep_digit([1, 2, 3]):
    ...     print(i)
    ...
    11
    22
    33
    """
    """YOUR CODE HERE"""
```

Reorder These Lines
`yield _________`
`for i in __________:`
`if _________:`



7. (18 points) 88's Bizarre Social Network

You are making a social network where users can make profiles with their name and age. Users can keep track of their friends by adding them to their friends list. You decide that in order to make money you make users pay for premium accounts. Default profiles are limited to one friend, while premium accounts can have up to 1000. In addition, premium accounts have the ability to leave comments on other user's profiles. Whenever a profile is created it is added to a list that contains all profiles ever created called 'all_profiles. Fill out the respective Profile and Premium classes to create your social network. Use the doctests as a guide for filling out the contents of each function.

The number of lines in each function is a guide. This is recommendation for a simple implementation, but we will not be enforcing line length.

```
class Profile():
    """
    >>> joseph = Profile("Joseph", 19)
    >>> jonathan = Profile("Jonathan", 20)
    >>> dio = Premium("Dio", 21) # Dio has a premium account
    >>> dio.name #profiles have an associated name
    ‘‘Dio’’
    >>> dio.age #profiles have an associated age
    21
    >>> len(Profile.all_profiles) # all_profiles should contain all three profiles: joseph, jonathan,
    3

    >>> joseph.add_friend(jonathan)
    >>> len(joseph.friends)
    1
    >>> joseph.add_friend(dio) # default profiles can only have one friend, so adding a second friend
    AssertionError: Friends list full!

    >>> dio.add_friend(joseph)
    >>> dio.add_friend(jonathan)
    >>> len(dio.friends) # premium profiles can have up to 1000 friends
    2

    >>> dio.comment(jonathan, "Oh? You're Approaching Me?")
    >>> jonathan.comments # premium profiles can leave comments on any profile. Comments are recorded
    ["Oh? You're Approaching Me?"]
    >>> jonathan.comment(dio, "What?") # Default accounts cannot comment and instead should print Need
    AssertionError: Need premium account to comment
    """
    all_profiles = []
    limit = 1

    def __init__(self, name, age):
        ----------------------------------
        ----------------------------------
        ----------------------------------
        ----------------------------------
        ----------------------------------

    def add_friend(self, friend):
        ----------------------------------
        ----------------------------------
        ----------------------------------
```

```
    def comment(self, account, msg):
        ----------------------------------

class Premium(Profile):
    def __init__(self, name, age):
        ----------------------------------
        ----------------------------------

    def comment(self, account, msg):
        ----------------------------------
```

(a) (6 pt)

Complete `Profile.__init__`

```
class Profile():
    """Review the doctests above"""
    def __init__(self, name, age):
        ----------------------------------
        ----------------------------------
        ----------------------------------
        ----------------------------------
        ----------------------------------
```

(b) (5 pt)

Complete `Profile.add_friend`

```
class Profile():
    """Review the doctests above"""
    def add_friend(self, friend):
        ----------------------------------
        ----------------------------------
        ----------------------------------
```

(c) (2 pt)

Complete `Profile.comment`

```
class Profile():
    """Refer to the doctests above"""
    def comment(self, account, msg):
        ----------------------------------
```

(d) (3 pt)

Complete `Premium.__init__`

```
class Premium(Profile):
    def __init__(self, name, age):
        ------------------------------------
        ------------------------------------
```

(e) (2 pt)

Complete `Premium.comment`

```
class Premium(Profile):
    def comment(self, account, msg):
        ----------------------------------
```

8. (20 points) Perfect SymmeTREE

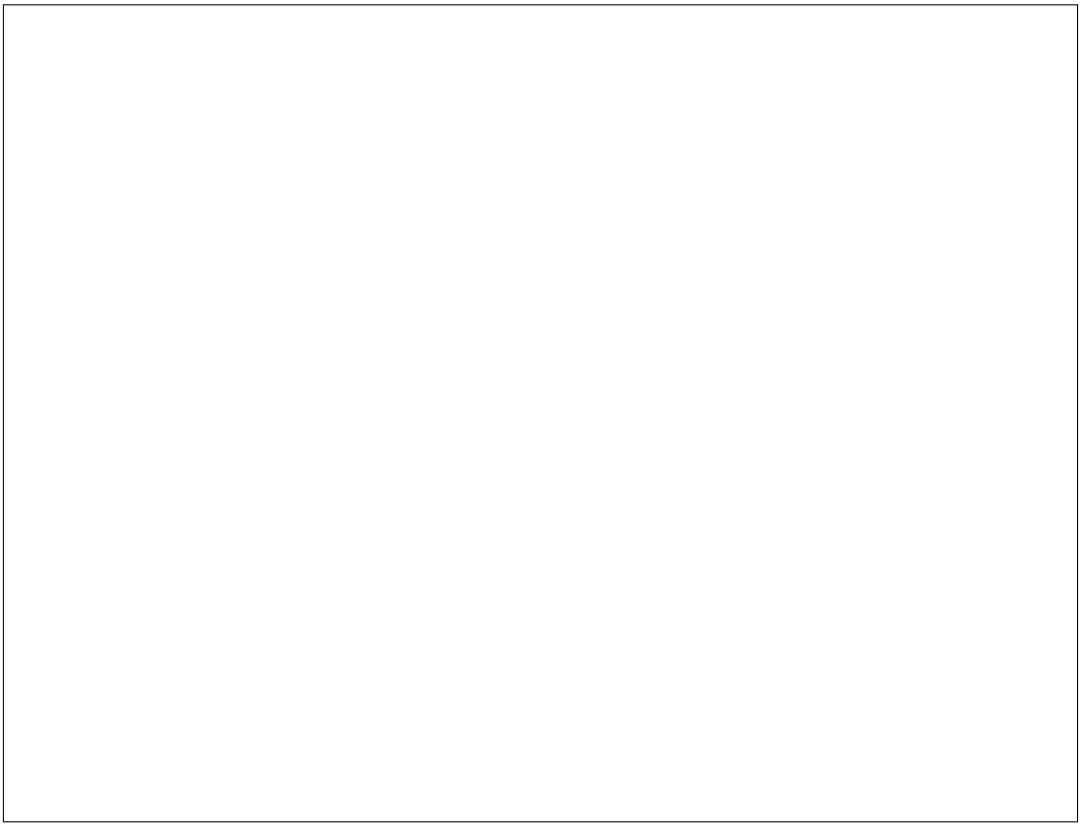
(a) (8 pt) We will call a symmetric decreasing tree a tree where every node has 2 branches and the values in every node decrease by 1 every time the depth increases by 1. Complete the function `symmetric_decreasing_tree` such that given an input `n`, it constructs a symmetric decreasing tree where the node at depth 0 (the top) has a value n, every node at depth 1 has a value `n - 1`, and so on, and all the leaves of the tree should have value 1. Assume that n >= 1.

Use the following lines of code as the bases for your solution. The following lines of code are provided out of order. You may need to reorder the lines and fill in the blanks. A reasonable solution exists which uses all line, but there may be a solution which does not use all lines. There may be multiple correct solutions. The length of blank inputs has *no meaning on what expressions might fill in those blanks.*

```
def symmetric_decreasing_tree(n):
    """
    >>> symmetric_decreasing_tree(3)
    Tree(3, [Tree(2, [Tree(1), Tree(1)]), Tree(2, [Tree(1), Tree(1)])])
    >>> symmetric_decreasing_tree(1)
    Tree(1)
    """
    """YOUR CODE HERE"""
```

Reorder the following lines

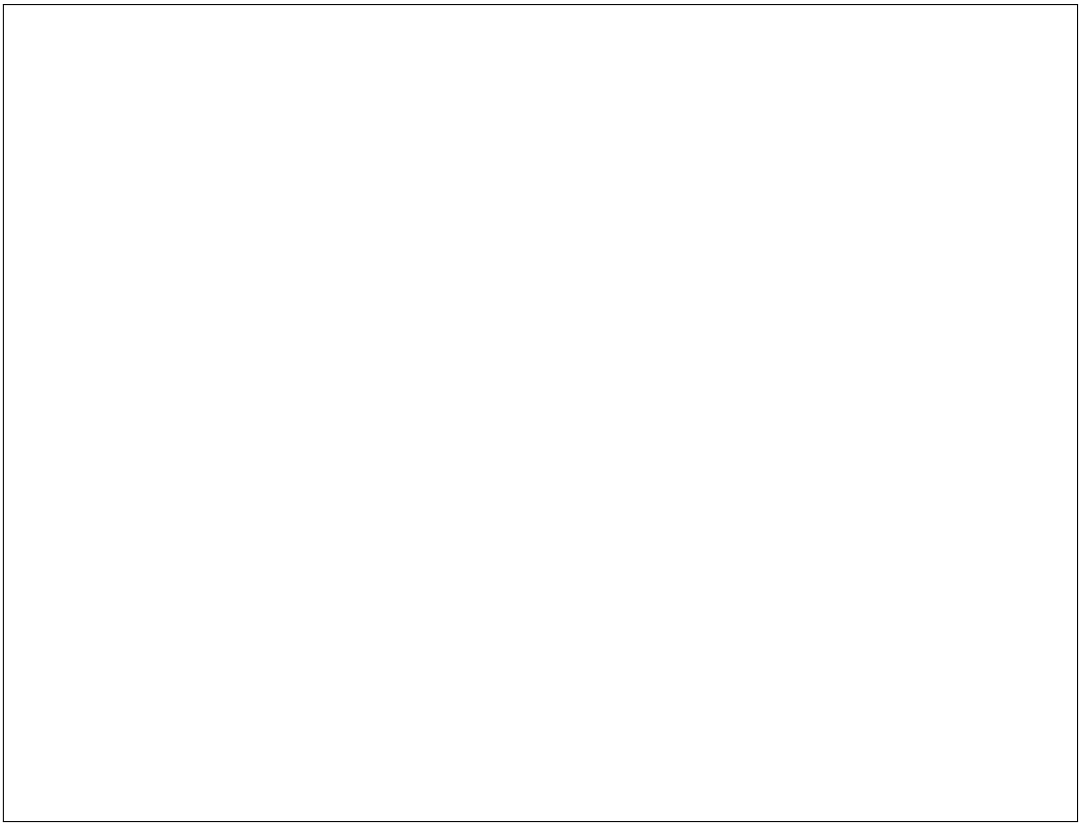
```
return Tree(_____________)
else:
if n == _______:
return Tree(________)
branches = ___________________________
for i in _________________:
___________ symmetric_drecreasing_tree(________) ________
```



(b) (12 pt) A 'tree'cherous villain wants to ruin our symmetric decreasing tree by removing valuable information from it and polluting the tree with extra unlucky nodes. The villain has provided a linked list of numbers he wants to erase from the tree. Help the villain finish this task by completing the following methods.

Implement a helper function `contains_val` that returns whether or not a linked list contains a value.

```
def contains_val(lnk, val):
    """
    >>> contains_val(Link(4, Link(7, Link(1, Link(3)))), 7)
    True
    >>> contains_val(Link(5, Link(6, Link(8))), 3)
    False
    """
    """YOUR CODE HERE"""
```



(c) (4 pt) Implement the function `destroy_tree` by changing any value in the tree that appears in the supplied linked list to 0. You may use the method implemented in the previous part.

Use the following lines of code as the bases for your solution. The following lines of code are provided out of order. You may need to reorder the lines and fill in the blanks. A reasonable solution exists which uses all line, but there may be a solution which does not use all lines. There may be multiple correct solutions. The length of blank inputs has *no meaning on what expressions might fill in those blanks.*

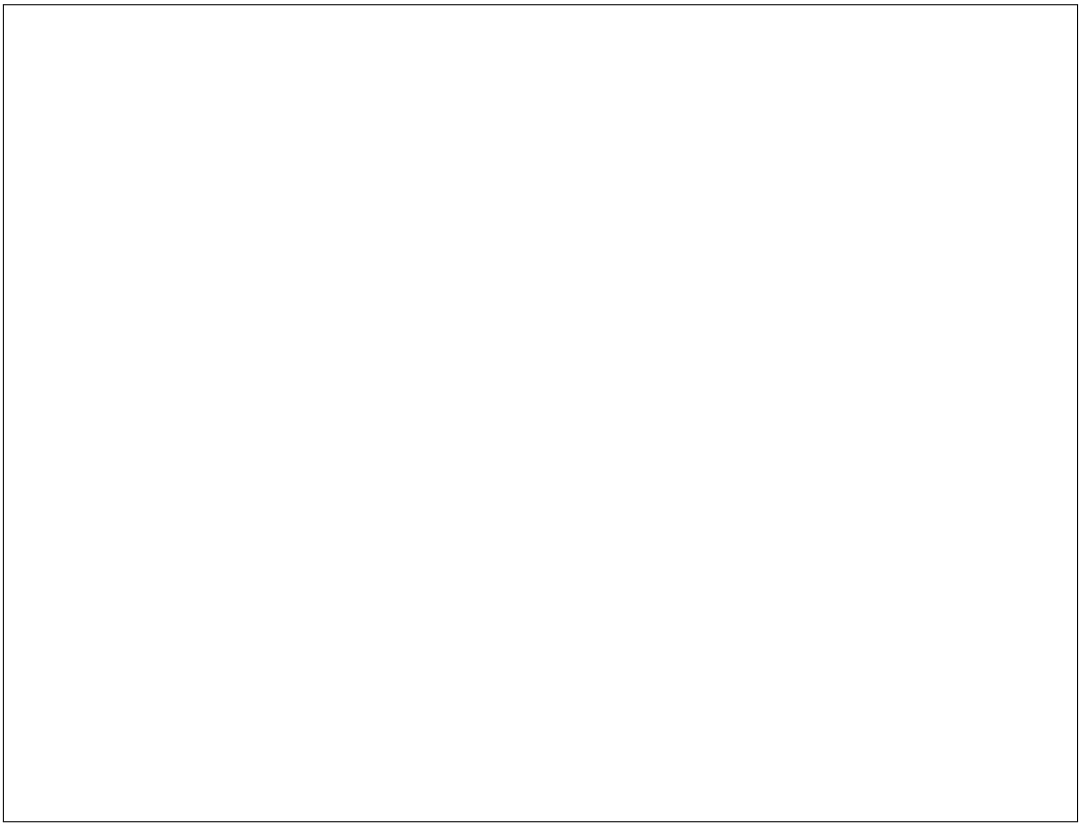
```
def destroy_tree(t, lnk):
    """
    >>> t3 = symmetric_decreasing_tree(3)
    >>> bad_nodes = Link(3, Link(5, Link(1)))
    >>> destroy_tree(t3, bad_nodes)
    >>> t3
    Tree(0, [Tree(2, [Tree(0), Tree(0)]), Tree(2, [Tree(0), Tree(0)])])
    """
    """YOUR CODE HERE"""
```

Reorder the following lines

```
for b in _______________:
_________(b, lnk)
t._________ = 0
if ________(______, ______):
```

(d) (4 pt) Implement the function add_unlucky_leaves that, given a list of unlucky numbers, mutates the original tree by adding all of these numbers as branches to each of the original leaves in the tree. You can assume the list of unlucky numbers is not a nested list.

```
def add_unlucky_leaves(t, unlucky_nums):
    """
    >>> t = Tree(0, [Tree(2, [Tree(1), Tree(5)])])
    >>> add_unlucky_leaves(t, [13, 6, 17])
    >>> t
    Tree(0, [Tree(2, [Tree(1, [Tree(13), Tree(6), Tree(17)]), Tree(5, [Tree(13), Tree(6), Tree(17)
    """
    """YOUR CODE HERE"""
```



No more questions.