ICS 31 • UC IRVINE
WINTER 2016 • DAVID G. KAY

YOUR NAME ______________________________
YOUR STUDENT ID (8 DIGITS) ________________
YOUR UCINET ID ___________________________

YOUR LAB SECTION (CIRCLE ONE):

1.	8:00a	Shibani Konchady
2.	10:00	Shibani Konchady
3.	4:00	Akshat Amrish Patel
4.	6:00	Akshat Amrish Patel
5.	8:00a	Aniket Shivam
6.	10:00	Aniket Shivam
7.	4:00	Roeland Singer-Heinze
8.	6:00	Roeland Singer-Heinze

First Midterm

You have 75 minutes (until the end of the class period) to complete this exam. There are 65 points possible, so allow approximately one minute per point and you'll have plenty of time left over.

Please read all the problems carefully. If you have a question on what a problem means or what it calls for, ask us. Unless a problem specifically asks about errors, you should assume that each problem is correct and solvable; ask us if you believe otherwise.

In answering these questions, you may use any Python 3 features we have covered in class, in the text, in the lab assignments, or earlier on the exam, unless a problem says otherwise. Use more advanced features at your own risk; you must use them correctly. If a question asks for a single item (e.g., one word, identifier, or constant), supplying more than one will probably not receive credit.

Remember, stay cool! If you run into trouble on a problem, go on to the next one. Later on, you can go back if you have time. Don't let yourself get stuck on any one problem.

You may not share with or receive from anyone besides the instructor or TAs any information or materials during the exam. You may not use any electronic devices.

Please write your answers clearly and neatly—we can't give you credit if we can't decipher what you've written.

We'll give partial credit for partially correct answers, so writing something is better than writing nothing. But be sure to answer just what the question asks.

Good luck!

Problem 1 (11 points)	
Problem 2 (9 points)	
Problem 3 (19 points)	
Problem 4 (5 points)	
Problem 5 (6 points)	
Problem 6 (6 points)	
Problem 7 (9 points)	
Total (65 points)	

Problem 1 (11 points)

Use the following definitions in this problem:

```
THIS_MONTH = "February"
DAYS = 29
holidays = ['King', 'Presidents', 'Ground Hog', "Valentine's"]
```

(a) (4 points) What does Python print as it executes the following sequence of statements? (Write your answers in the blank space to the right of the code.)

```
print("Let's Go!")
print(DAYS * 100)
print(THIS_MONTH, DAYS, "is Leap Day")
print(len(THIS_MONTH) < 10)
```

(b) (3 points) What does Python print as it executes the following sequence of statements? (Write your answers in the blank space to the right of the code.) Remember zero-based indexing.

```
print(holidays[1])
print(holidays[0] == 'KING')
print(holidays[2], 'Day')
print(holidays[2] + 'warts')
print("Valentine's" in holidays)
print(holidays[0]*3)
```

(c) (4 points) What does Python print as it executes the following sequence of statements? (Write your answers in the blank space to the right of the code.) Remember zero-based indexing.

```
print(holidays[-1], holidays[1])
print(len(holidays[1:]), len(holidays))
print(holidays[1], holidays[2], holidays[3])
print("Ground" in holidays, "Ground" in holidays[2])
```

Problem 2 (9 points)

The Anteater Bakery represents each kind of bread it sells in a namedtuple called Loaf that has four fields: the kind of loaf (like "whole wheat" or "seeded rye"), its weight in ounces, its price, and the number currently in stock.

(a) (3 points) Which of the following defines a namedtuple that satisfies this specification? Circle *one or more* of A, B, C, D, E, F, or G; more than one may be correct.

A. `Loaf = namedtuple('Loaf', 'type size cost quantity')`
B. `Whole_Wheat = Whole_Wheat('name ounces dollars number')`
C. `Loaf = namedtuple('Loaf', 'Baguette', 14, 2.75, 50)`
D. `Loaf = namedtuple('Loaf', 'kind weight price instock')`
E. `Loaf = namedtuple('Loaf', 'Sourdough, 15, 2.50, 75')`
F. `Loaf = namedtuple('Loaf', 'name ounces dollars quantity')`
G. `Loaf = Loaf('Challah', 20, 5.50, 25)`

(b) (3 points) Which of the following creates a new Loaf object, following the definition above, to represent a stock of 40 loaves of multigrain bread, each weighing 24 ounces and costing $7.50? Circle *one or more* of A, B, C, D, E, or F; more than one may be correct.

```
A. new_loaf = Loaf.Multigrain(24, 7.50, 40)
B. new_loaf = Loaf('Multigrain, 24, 7.50, 40')
C. new_loaf = Loaf('Multigrain', 24, 7.50, 40)
D. new_loaf = Loaf('Multigrain', 24, 7.50, 40, 'sliced')
E. new_loaf = namedtuple('Loaf', 'Multigrain 24 7.50 40')
F. new_loaf = Loaf(40, 'Multigrain', 24, 7.50)
```

(c) (3 points) You want to reduce the price of each loaf by $0.50 to celebrate the bakery's 50th anniversary. Which of the following statements correctly decreases the price of the Loaf object stored in `a_loaf` (according to at least one of the correct definitions in part (a))? Circle *one or more* of A, B, C, D, E, or F; more than one may be correct.

```
A. a_loaf = a_loaf._replace(price = a_loaf.price - 0.50)
B. a_loaf.price = a_loaf.price - 0.50
C. a_loaf = a_loaf._replace(a_loaf.price = 7.00)
D. a_loaf = Loaf(a_loaf.kind, a_loaf.weight, a_loaf.price - 0.50, a_loaf.instock)
E. a_loaf = Loaf(price = a_loaf.price - 50)
F. print(a_loaf.price - 0.50)
```

Problem 3 (19 points)

The Anteater Bakery has branches worldwide. Their management has asked you to analyze a recent customer satisfaction survey. Customers at every branch have given the branch a 0–100 score in three categories: quality of the bread, the service, and the ambience (physical environment). You use this namedtuple to represent a branch's survey results:

```
Branch = namedtuple('Branch', 'location ID bread service ambience')
```

The location is a string, the ID is an int, and the remaining fields are floats storing the average rating from all customers on each category for that branch.

(a) (4 points) In the function definition below, fill in each blank with a single Python constant, operator, or identifier name (variable, function, attribute, method) to satisfy the problem specification.

```
def overall_satisfaction(b: Branch) -> float:
    """ Return the branch's overall satisfaction score, in the range 0-100, computed
        as an equally-weighted average of its bread, service, and ambience scores.
    """
    return (__________.bread __________ b.service + b.__________) / __________
```

(b) (1 point) Fill in each bank in the assert statements below so that each assertion will be true (assuming a correct solution to part (a)).

```
assert overall_satisfaction(Branch('Santa Ana', 111, 25, 50, 75)) == _____

assert overall_satisfaction(Branch('Corona del Mar', 222, _____, _____, _____)) == 100
```

(c) (6 points) What do the following statements print? [You can do the arithmetic in your head.]

```
B1 = Branch('Orange', 333, 85, 90, 95)
B2 = Branch('Newport Beach', 444, 97, 97, 97)
print(B1.location, "(#", B1.ID, '):', overall_satisfaction(B1), 'overall/100')
print(B2.location, "(#", B2.ID, '):', overall_satisfaction(B2), 'overall/100')
```

(d) (2 points) In the function definition below, fill in each blank with a single Python constant, operator, or identifier name (variable, function, attribute, method) to satisfy the problem specification.

```
def Branch_bread(b: Branch) -> float:
    ''' Return a Branch's score for bread
    '''
    return __________ . __________
```

(e) (6 points) In addition to the function `Branch_bread` defined above, assume you have also defined similar functions called `Branch_service` and `Branch_ambience` that return the values of those fields. Also, suppose that Anteater Bakery has a list of 1750 Branches called `BL`.

(e.1) Fill in the blanks below to reorder `BL` by each Branch's ambience score, lowest to highest:

```
_______________.sort(key=_______________)
```

(e.2) Fill in the blanks below to reorder `BL` by each Branch's overall satisfaction score, lowest to highest. You may use other functions that have been defined in this exam.

```
_______________.sort(key=_______________)
```

(e.3) Write a single Python statement similar to those in (e.1) and (e.2) to reorder BL by each Branch's overall satisfaction score, highest to lowest. You may use other functions that have been defined in this exam.

Problem 4 (5 points)

In the function definition below, fill in each blank with a single Python constant, operator, or identifier name (variable, function, attribute, method) to satisfy the problem specification.

```
def bread_higher (branches: 'list of Branch') -> 'list of Branch':
    """ Return a list of those Branches on the specified list whose bread score is
         higher than its service score
    """
    result = [ ]
    for b in ____________:
        if b.____________  ____________ b.service:
            ____________.append(____________)
    return result
```

Problem 5 (6 points)

Suppose we have a list of Branches called `BL`, as in the previous problem. Fill in each blank with one of the ten code segments below (A through J) to match its most accurate description. Don't use any segment (A–J) more than once.

A.
```
for b in BL:
    print(b.location, b.overall_satisfaction)
```

B.
```
for b in BL:
    if b.ambience >= 85:
        print(b)
```

C.
```
for b in BL:
    if overall_satisfaction(b) > 85:
        print(b.location, b.ID)
```

D.
```
for b in BL:
    print(b.location, overall_satisfaction(b))
```

E.
```
for b in sorted(BL, key=overall_satisfaction, reverse=True):
    print(b.location, b.ID)
```

F.
```
for b in BL:
    if b.ambience >= 85:
        print(b.location)
```

G.
```
for b in BL:
    print(b)
```

H.
```
for b in BL:
    print(BL[b].location, overall_satisfaction(BL[b]))
```

I.
```
for b in BL:
    print(b.location, overall_satisfaction(b) > 85)
```

J.
```
for b in BL:
    print(b.location)
```

___ Prints the location and overall satisfaction score for every Branch

___ Prints the locations of all Branches whose ambience is at least 85

___ Produces an error message about the improper use of `overall_satisfaction`

___ Prints the location and ID of each Branch with overall satisfaction over 85

___ Prints all the information about each Branch in Python namedtuple form

___ Prints the location of each Branch, one per line

Problem 6 (6 points)

Identify the data type of each of the following expressions, using definitions that appear in this exam.

Chose from: `int` `float` `bool` `str` `Branch` list of `Branch` list of `str` list of `float` function

```
holidays[2:]

len(holidays[3:])

DAYS

holidays[0]

B2

B2.bread

[B2, B1, B1, B2]

THIS_MONTH[1] == 'K'

Branch_bread

'Hog' in holidays

overall_satisfaction(B1)

THIS_MONTH[0:3] + '!'
```

Problem 7 (9 points)

What does the following code print out?

```
def n_copies(n: int, s: str) -> str:
    return s * n

print("Almost done!")
a = n_copies(2, 'cat')
print(a)
print("More:", n_copies(3, a))
print(n_copies(2, n_copies(4, 'O')))
print("The finish line.")
```

Problem 8 (0 points)

When you're done with the exam, follow these steps (so you don't disturb your classmates and so your exam gets turned in properly):

- Gather up all your stuff.
- Take your stuff and your exam down to the front of the room.
- Turn in your exam; show your ID if asked.
- Exit by the doors at the front of the room. Don't go back to your seat or disturb students still working.