

Name: \_\_\_\_\_

**Rules and Hints**

- You may use one handwritten 8.5 x 11" cheat sheet (front and back). This is the only additional resource you may consult during this exam. *No calculators.*
- When you write code, be sure that the indentation level of each statement is clear.
- Explain/show work if you want to receive partial credit for wrong answers.
- As long as your code is correct, you will get full credit. No points for style.
- As always, the SSU rules on academic integrity are in effect.

<b>Problem</b>	<b>Max Score</b>	<b>Your Score</b>
<i>Problem 1: Trace Code</i>	40	
<i>Problem 2: Write snippets of code</i>	30	
<i>Problem 3: Write a complete program</i>	30	
<b>Total</b>	100	

**Problem 1: Trace Code (40 points)**

Write what will be printed to the screen when each of the following snippets of code is executed in PyCharm or in the Online Python Tutor.

Be very clear with spacing, line breaks, etc.

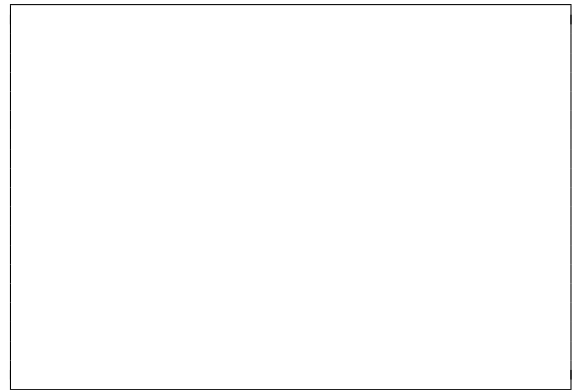
Write your final answer clearly in each box.

Treat each sub-problem as an independent question.

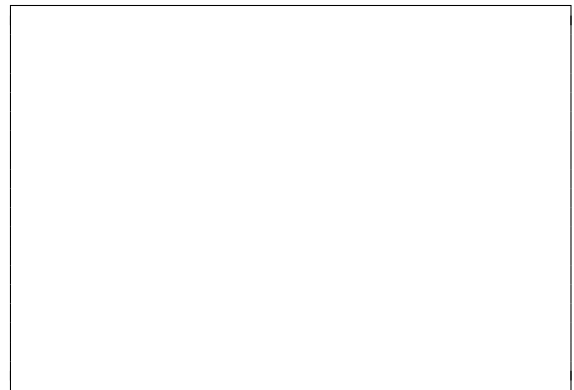
All questions in this section are worth 5 points.

**Problem 1A**

```
prizes = 4
print(1 + 10*prizes)
```

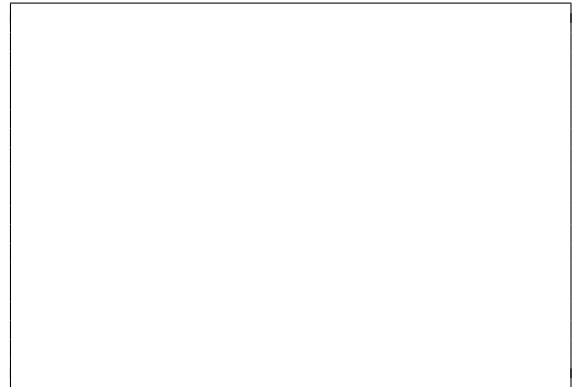
**Problem 1B**

```
z = 3 ** 2
print(z)
z = z / 3
print(z)
```

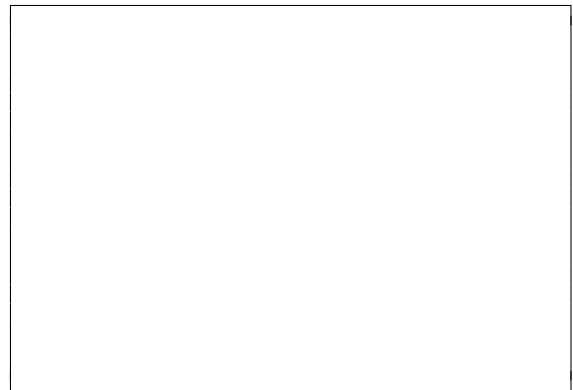


**Problem 1C**

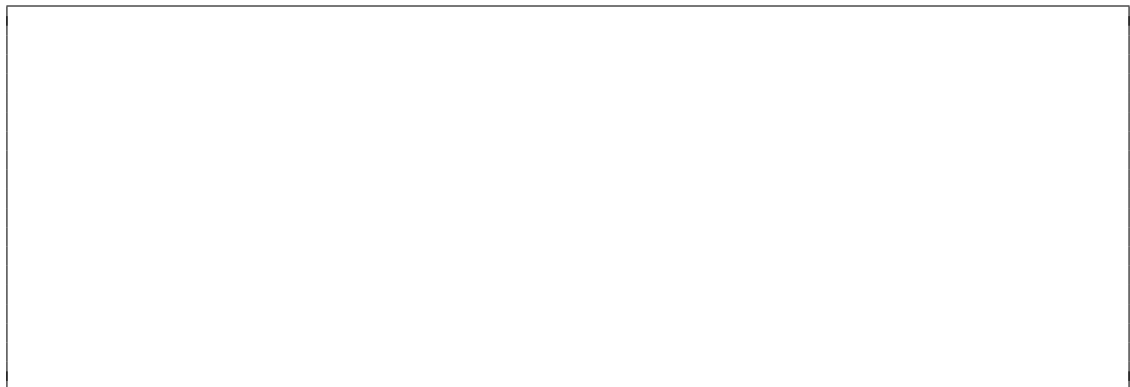
```
a = 8
b = 19
c = b // a
d = b % a
print(c,d)
```

**Problem 1D**

```
bob = 'cat'
print(bob, 'cat')
print('bob' + bob)
```

**Problem 1E**

```
print('Lobo ', 'Deal ', end='! ')
print('Loaded Nachos ', 'each ', '2.99 ', sep='$')
```



**Problem 1F**

```
x = 45
print("Bananas to ship:")
if (x > 1):
    print("more than one")
if (x > 100):
    print(x, "hundreds")
elif (15 < x < 100):
    print("bunches")
else:
    print("many")
```

**Problem 1G**

```
k = 10
for j in range(10,13):
    print(j)
    k = k + j
print(k)
```

**Problem 1H**

```
for k in range(3,6):
    for j in range(2):
        print(k-j, end=' ')
    print('!!')
```

**Problem 2: Write snippets of code (30 points)**

Write snippets of code to do the following. Your code should only print the requested output. You will lose points for printing additional output.

You can assume that all your snippets are enclosed within a main function and that any necessary libraries have been imported. You only need to write the specific lines of code that accomplish each task.

**Problem 2A (8 points)**

Prompt the user to enter a temperature in Fahrenheit, then calculate and output that temperature in Kelvin. Use the following formula to convert Fahrenheit ( $^{\circ}\text{F}$ ) to Kelvin (K):

$$K = \frac{5}{9}(F + 459.67)$$

For example, 47.5 degrees Fahrenheit is 281.7611 Kelvin.

**Problem 2B** (10 points)

Ask the user to enter their favorite fruit.

If the user types “prune” or “raisin” or “sultana”, remark on their choice by saying “Dried fruits are nice.”

If the user types “eggplant” or “cucumber”, remark by saying “Technically, a berry: clever you!”

For any other fruit, remark by saying “Delicious.”

**Problem 2C** (12 points)

Ask the user to enter 200 integers, one at a time. After the user has entered all the integers, print the sum of all the *negative integers*, print the *minimum* integer, and print the *sum of the positive even* integers. You may assume the user enters at least one positive integer and at least one negative integer.

For example, if the sequence 5, -2, 4, -15, 6, 10 was entered, the program should first output -17 (because the sum of -2 and -15 is -17), then output -15 (the smallest integer entered), and then output 20 (since 4, 6, 10 are the positive evens and their sum is 20).

**Problem 3: Write a complete program (30 points)**

For this problem, you must write a complete program. This includes writing a docstring, logic in `def main()`, a call to `main()`, any necessary library imports, etc.

Read the instructions carefully before you start coding! If you get stuck, try to maximize your partial credit. Your program is a diary that gathers data about dogs boarded at a veterinarian. The program should do the following:

- Prompt the user to enter the total number of dogs at the vet.
- For each dog:
  - Prompt for the name of the dog, the weight of the dog (in pounds), the amount of kibble they ate (in pounds) and the amount of water they drank (in gallons), matching the example prompts below *exactly*.
  - If the amount of water the dog drank was less than  $0.004w$  (where  $w$  is their weight), then output “Watch for dehydration.”
- After the user has entered all of their data:
  - Print the average water each dog consumed.
  - Print the number of dogs that should be watched for dehydration.
  - Print the name of the dog who ate the most and how much kibble they ate.

Your program should match the sample output below, *exactly*. User input is underlined.

```
* Dog Diet Diary *
Enter number of dogs: 2

Dog 1: Ruffles
Weight (lbs): 4.1
Kibble (lbs): 1.2
Water (gals): 0.55

Dog 2: Sunny
Weight (lbs): 65.3
Kibble (lbs): 1.1
Water (gals): 0.2
-> Watch for dehydration.

* Summary *
Average water per dog: 0.375 gals
Dogs to be watched: 1
Ruffles ate the most (1.2 lbs of kibble)
```



**Problem 3, continued ...**

**Extra Page**