

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:</i> Trace Code	20	
<i>Problem 2:</i> Trace Functions	20	
<i>Problem 3:</i> Define functions	30	
<i>Problem 4:</i> Write a complete program	30	
<b>Total</b>	100	

**Problem 1: Trace Code (20 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.

Write your final solution in the box provided

Do not write any scratch-work in the solution box.

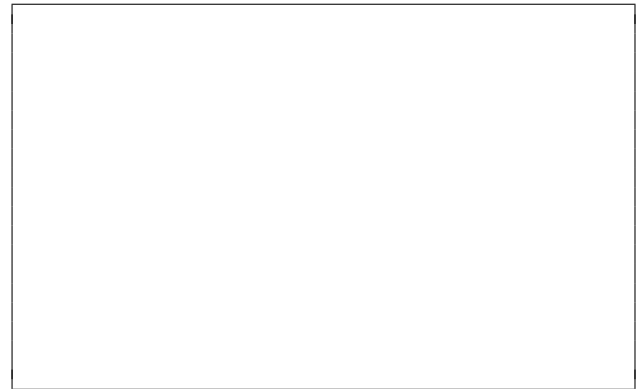
In your solution, be very precise with spacing, line breaks, etc.

Treat each sub-problem as an independent question.

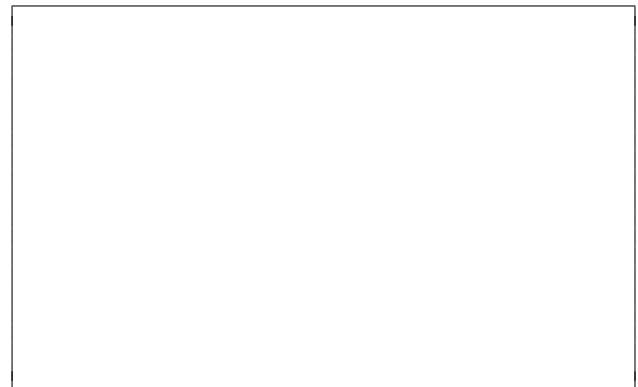
All questions in this section are worth 4 points.

**Problem 1A**

```
x = 1
y = 6
while x < y:
    x = x+2
    print(x, y)
```

**Problem 1B**

```
s = 'Go Seawolves!'
print(len(s))
print(s[len(s)-1])
print(s[:5])
print(s[-7:-1])
```



**Problem 1C**

```
s = ['lion', 'tiger',  
     'bear', 'oh', 'my']  
print(len(s))  
print(s[2])  
print(len(s[2]))  
print(s[2][1])
```

**Problem 1D**

```
A = [0,2,4,6,8]  
B = A  
C = A[:]  
B[0] = sum(A)  
C[2] = sum(A)  
print(B)  
print(C)
```

**Problem 1E**

```
L = [1,2,3]  
try:  
    print('L =', L)  
    L = L[:-1] + 4  
    print('L =', L)  
except Exception as e:  
    print('error:', L)
```

**Problem 2: Trace Functions (20 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.

Write your solution in the box provided.

Do not write any scratch-work in the solution box.

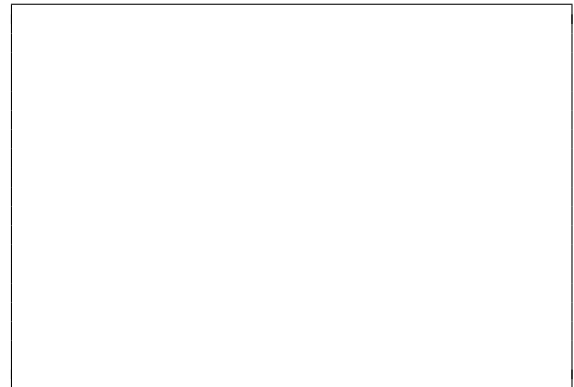
In your solution, be very precise with spacing, line breaks, etc.

Treat each sub-problem as an independent question.

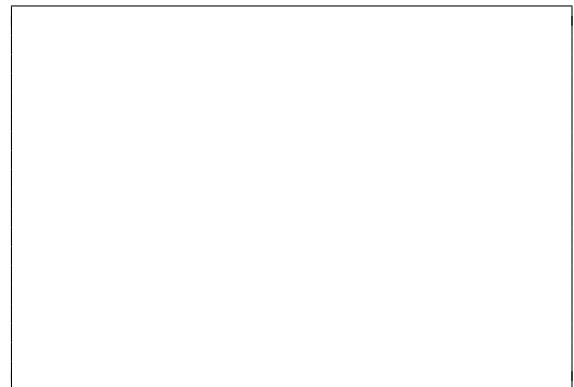
All questions in this section are worth 5 points.

**Problem 2A**

```
def cat(a,b):  
    return a+b  
  
a = 'xxx'  
b = 'yyy'  
z = cat(b,a)  
print(z)
```

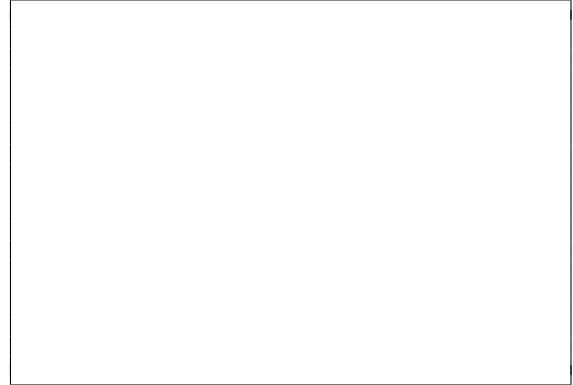
**Problem 2B**

```
def mystery(str):  
    s = ''  
    for c in str:  
        s = c + s + c  
    return s  
  
str = mystery('ba')  
print(str)
```



**Problem 2C**

```
def combineFirstLast(A):  
    return A[0] + A[-1]  
  
A = [[1], [2,3], [4,5,6]]  
L = combineFirstLast(A)  
print(len(A), len(L), L)
```

**Problem 2D**

```
def subl(L):  
    for i in range(len(L)):  
        L[i] = L[i] - L[0]  
  
A = [1,2,3]  
subl(A)  
print(A)
```



**Problem 3: Define functions (30 points)**

Define functions to perform the following tasks, obeying the stated requirements, including:

- Assume that the math library has been imported for you.
- Do not ask the user for input unless the specification explicitly requires it.
- Do not print anything unless the specification explicitly requires it.
- Do not call any function unless the specification explicitly requires it.

**Problem 3A (10 points)**

Write a function named `area` that enables this code:

```
print("Triangle with b", 10, "and h", 20, "has area", 10*20/2)
print("Triangle with b", 15, "and h", 11, "has area", 15*11/2)
print("Triangle with b", 7, "and h", 5, "has area", 7*5/2)
```

to be replaced with the following code:

```
area(10,20)
area(15,11)
area(7,5)
```

**Problem 3B** (10 points)

Define a function named `word_reverse` that:

- Takes 1 parameter: a string
- Returns a string with the words in reverse order

Example of calling this function:

```
word_reverse("The quick brown fox") returns "fox brown quick The"
```

**Problem 3C** (10 points)

Define a function named `getRowSum` that:

- Takes 2 parameters: a nested (2-dimensional) list and an integer `j`
- Returns the sum of all elements in row `j`

You can assume that row `j` exists.

Example of calling this function:

Let `L = [[5,10,15], [1,2,3], [7,8,9]]`, then  
`getRowSum(L,0)` returns 30



**Problem 4: 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. To get full credit, your functions should call each other where appropriate and avoid duplicating code. In all examples, user input is underlined.

Your program should contain the following functions:

- `get_names`:
  - Takes no parameters. Prompts the user for names until the user enters a blank line. Each line of user input should be considered a name. Returns a list of the names the user entered (not including the blank line).

Example: A call to `get_names` with the below sample user input

```
Enter a name: Harry James Potter
Enter a name: Hermione Jean Granger
Enter a name:
```

should return the list `['Harry James Potter', 'Hermione Jean Granger']`.

- `make_initials`:
  - Takes a string as a parameter and returns a string that holds its initials (the first letter from each word in the original string, in upper case).

Example: Given `'harry james Potter'`, the function returns `'HJP'`.

- `main`:
  - Calls `get_names` to get a list of names from the user. For each name in the list, creates the initials by calling `make_initials` and prints the name and initials for it. Given the sample inputs below, your program should match the sample output exactly.

Sample input/output:

```
Enter a name: Harry James Potter
Enter a name: Hermione Granger
Enter a name: Ronald Bilius weasley
Enter a name:
```

```
Harry James Potter (HJP)
Hermione Granger (HG)
Ronald Bilius weasley (RBW)
```

*Start your solution on the next page...*

**Problem 4, continued ...**

**Problem 4, continued ...**