

Name: _____

Rules and Hints

- You may use one handwritten 8.5×11 " cheat sheet (front and back). This is the only additional resource you may consult during this exam. *No calculators.*
- 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.
- When you write code, be sure that the indentation level of each statement is clear.

Grade

Problem	Your Score	Max Score
<i>Problem 1:</i> Trace snippets of code		15
<i>Problem 2:</i> Trace functions		15
<i>Problem 3:</i> Define functions		30
<i>Problem 4:</i> Write a complete program		40
Total		100

Problem 1: Trace snippets of code (15 points)

What will print 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.

Treat each sub-problem as an independent question.

Problem 1A

```
x=8
while x > 4:
    x=x-2
    print(x)
```

Problem 1B

```
s = 'Tales of Legendia'
print(s[3:8])
print(s[-3:-1])
print(s.upper())
```

Problem 1C

```
strings = ['Lilo', 'Stitch', 'Experiment626', 'Animation']
print(len(strings))
print(strings[2])
print(strings[2][2])
```

Problem 2: Trace functions (15 points)

What will print to the screen when each of the following snippets of code is executed in PyCharm?

Be very clear with spacing, line breaks, etc.

Note: the parts of this problem are independent.

Problem 2A

```
def calculate(w, x, y):  
    a = x  
    b = w + 1  
    return a + b + w  
  
print(calculate(1, 2, 0))
```

Problem 2B

```
def add_one(x):  
    y = x + 1  
    return y  
  
x = 2  
x = x + add_one(x)  
print(x)
```

Problem 2C

```
def inc(B):  
    for i in range(len(B)):  
        B[i] = B[i] + 1  
  
A = [2,6,7]  
inc(A)  
print(A)
```

Problem 3: Define functions (30 points)

Define functions to perform the following tasks, obeying these requirements:

- 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.

Problem 3A

Consider this code

```
print("The area of circle with radius 5 is ", math.pi * 5 ** 2)
print("The area of circle with radius 8 is ", math.pi * 8 ** 2)
print("The area of circle with radius 2 is ", math.pi * 2 ** 2)
```

Eliminate the redundancy in this code by defining an `area` function that takes 1 parameter: radius of circle and prints out the area of circle. This function can be then called three times to produce exactly the same output as above:

```
area(5)
area(8)
area(2)
```

Problem 3B

Define a function named `num_evens` that:

- Takes 1 parameter: a list of integers
- Returns the number of even integers in that list

Problem 3C

Define a function named `columnSums` that:

- Takes 1 parameter: a nested (2D) list of integers
- Returns a new list that contains the sum of each column in the input list. All sublists are of the same length.

For example, `columnSums([[5, 10, 15], [1, 2, 3]])` will return `[6, 12, 18]`

Problem 4: Write a complete program (40 points)

For this problem, you must write a complete program. That includes a docstring, a `def main()`, any necessary library imports, etc.

Each function you write should obey the rules given in Problem 3. **To get full credit, your functions should call each other where appropriate to avoid duplicating work!**

Your program should contain the following functions:

- `read_names`: Takes no parameters. Prompts the user for names until the user enters a blank line. Returns a list of the names the user entered (not including the blank line). Each line of user input should count as a single name, even if it has multiple words.

For example:

Enter a name: Don Draper

Enter a name: Peggy Olson

Enter a name:

should return the list ['Don Draper', 'Peggy Olson']

- `goodness`: Takes a string as a parameter and returns an integer that represents "goodness" of the string, which is determined as follows:
 - If the string contains one or more of letter "u", then goodness is 0
 - Otherwise, the goodness of a string is equal to the number of letter "g" in the string.

For example:

`goodness('Don Draper')` is 0 (Because there are no "g"s)

`goodness('Peggy Olson')` is 2 (Because there are 2 "g"s and no "u")

`goodness('Doug Wilson')` is 0 (Because there is one "u")

- `average`: Takes a list of integers as a parameter and returns their average.
- `main`: Calling your other functions whenever possible, this function should get a list of names from the user (stopping when the user enters a blank line). It should then print the following:
 - Each name followed by its "goodness" score.
 - All of the names whose "goodness" is greater than the average goodness computed from all names.

Problem 4 code

Problem 4, continued