

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.

Problem	Max Score	Your Score
<i>Problem 1: Trace Code</i>	20	
<i>Problem 2: Trace Functions</i>	20	
<i>Problem 3: Define functions</i>	30	
<i>Problem 4: Write a complete program</i>	30	
Total	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.

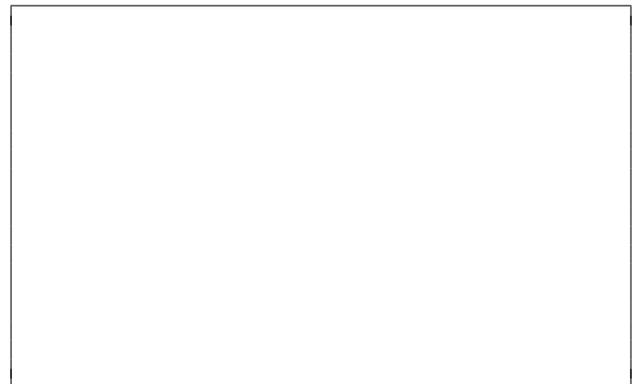
All questions in this section are worth 4 points.

Problem 1A

```
y = 1
while y < 10:
    y = y*3
    print(y)
```

**Problem 1B**

```
s = 'Spring Break '
print(len(s))
print(s[4:len(s)])
print(s[-10:-4])
print(s[:7]*3)
```



Problem 1C

```
s = ['It', 'Feels',  
    'Like', 'Summer',  
    'Of sixty-nine']  
print(len(s))  
print(len(s[3]))  
print(s[4])  
print(s[2][2])
```

Problem 1D

```
A = [-3, -2, 0, 2]  
B = A[:]  
C = A  
B[3] = -3  
C[1] = 2  
print(B)  
print(A+C)
```

Problem 1E

```
L = [1, 2, 3]  
i = -2  
while i < 1:  
    i = i + 1  
    L.append(i)  
    print(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 intdiv(m, n):  
    return m // n  
  
n = 25  
m = 7  
z = intdiv(n, m)  
print(z)
```

**Problem 2B**

```
def mystery(s):  
    str = ''  
    for char in s:  
        str = char*2 + str  
    return str  
  
answer = mystery('cs1')  
print(answer)
```

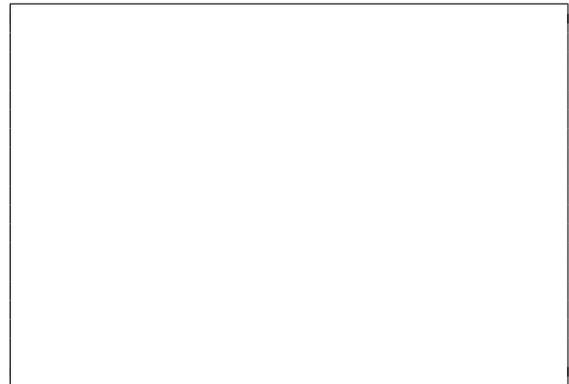


Problem 2C

```
def getElem(A):  
    return A[2+(len(A) % 2)]  
  
A = [0, 1, 2, 4, 8,  
     3, 6, 4, 8]  
La = getElem(A)  
  
B = [[0, 1], [2, 4, 8],  
     [3, 6], [4, 8]]  
Lb = getElem(B)  
  
print(La, Lb)
```

**Problem 2D**

```
def modifyList(L):  
    for e in L:  
        e = e + 3  
    print(L)  
  
    for i in range(len(L)):  
        L[i] = L[i]*3  
    print(L)  
  
A = [3,2,1]  
modifyList(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 (5 points)

Define a function named `bytes_to_megabytes` that:

- Takes 1 parameter: an integer representing bytes
- Returns a float representing the value in megabytes. Each megabyte holds 1,000,000 bytes.

For example, `bytes_to_megabytes(2500000)` returns 2.5

Problem 3B (10 points)

Define a function named `glue_under` that:

- Takes 1 parameter: a list of strings
- Returns a string that consists elements from the list concatenated together such that: the string starts with an underscore, contains every string from the list separated by underscores, and ends with an underscore.

Example of calling this function:

```
glue_under(['a', 'bc', 'def', 'g', 'hi']) returns '_a_bc_def_g_hi_'
```

Problem 3C (10 points)

Define a function named `getRowMax` that:

- Takes 2 parameters: a nested list (i.e., a 2-dimensional list) and an integer `i`
- Returns the maximum of all the elements in row `i`

You may assume that row `i` exists.

Example of calling this function:

Let `L = [[5,7,10,4], [3,2,1], [8,9], [0,6,5,2]]`, then

`getRowMax(L,3)` returns 6 because row 3 is `[0,6,5,2]` which has max value 6.

Problem 3D (5 points)

Define a function named `compareRows` that:

- Takes 3 parameters: a nested list (i.e., a 2-dimensional list) and two integers `i` and `j`
- Returns `True` if the maximum element in row `i` of the nested list is greater or equal to the maximum element in row `j` of the nested list; Otherwise Returns `False`
- Calls the function `getRowMax` (which was defined in Problem 3C) to compute maximum of an individual row

Example of calling this function:

Let `L = [[5,7,10,4], [3,2,1], [8,9], [0,6,5,2]]`, then

`compareRows(L,0, 2)` returns `True` because maximum element in row 0 (10) is greater than maximum element in row 2 (9)

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.

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

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

```
Enter a name: sesame st.
Enter a name: ernie
Enter a name: big BIRD
Enter a name:
```

should return the list `['sesame st.', 'ernie', 'big BIRD']`

- `make_proper`:
 - Takes 1 parameter: a string holding a noun. Returns a new string that is the same as the original string, except the first letter of each word is capitalized, all other letters are lower case, the final string does not start or end with any space, and intermediate whitespace is simplified.

Example: For `'sesame st.'`, the function returns `'Sesame St.'`

Example: For `'ernie'`, the function returns `'Ernie'`

Example: For `' big BIRD '`, the function returns `'Big Bird'`

- `main`:
 - Calls `get_nouns` to get a list of nouns from the user. Use `make_proper` to print the proper noun versions of each, printing *in reverse order*.

Sample input/output:

```
Enter a name: kErmit t. frog
Enter a name: BIG BIRD
Enter a name: bert and ernie
Enter a name:
```

```
Bert And Ernie
Big Bird
Kermit T. Frog
```

Start your solution on the next page...

Problem 4, continued ...

Problem 4, continued ...