CS 115 Exam 2 — Thu. 04/07/2016

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**
Problem 1: Trace snippets of code		20
Problem 2: Trace functions		20
Problem 3: Define functions		30
Problem 4: Write a complete program		30
Total		100

Problem 1: Trace snippets of code (20 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=1
while x < 7:
  print(x, end = ' ')
  x=x+2
```

Problem 1B

```
s = 'Mastermind'
print(len(s)])
print(s[0:len(s)])
print(s[3:-3])
```

Problem 1C

```
strings = ['Guess', 'Palette', 'Secret']
print(len(strings))
print(strings[1])
print(strings[1][1])
```

Problem 1D

```
A = ["of",1, 3]
B = A
C = A[:]
B[2] = "pirates"
C[1] = "caribbean"
print(A)
print(B+C)
```

Problem 2: Trace functions (20 points)

What will print to the screen when each of the following snippets of code is executed in PyCharm? Note: the parts of this problem are independent.

Problem 2A

```
def add_one(x):
    return x + 1

y = add_one(2)
print(y)
```

Problem 2B

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

z = calculate(1, 0, 2)
print(z)
```

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 2D

```
def add_work(x):
    y = "work is " + x
    return y

x = "Hard"
x = x + add_work(x)
print(x)
```

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.
- Do NOT call any function unless the specification explicitly requires it.

Problem 3A

Define a function named `countStars` that:

- Takes 1 parameter: a `string`
- Returns the number of times the character * appears in that `string`

Example of calling this function: `countStars("Sta**y Night")` will return `2` (because there are two * in that `string`)

Problem 3B

Define a function named `isPositive` that:

- Takes 1 parameter: a `list` of integers
- Returns `True` if all numbers in that `list` are positive; `False` otherwise

Example of calling this function: `isPositive([1,3,4,-1])` will return `False` (because there is one negative number in that `list`)

Problem 3C

Define a function named `rowSum` that:

- Takes 2 input parameters: a 2D `list` of integers `A` and `index i` of the row
- Returns the sum of row `i` in the input `list A`.

Example of calling this function: `rowSum( [ [5,10,15],[1,2,3] ], 1)` will return `6` (because `row 1` is `[1,2,3]` and sum of it's elements is `6`)

Problem 3D

Define a function named `getListSum` that:

- Takes 1 input parameter: a 2D `list` of integers `A`
- Returns a `list` containing the sum of each row in the input `list A`.
- Calls the function `rowSum` (which was defined in Problem 3C) to compute sum of individual rows

Example of calling this function: `getListSum( [ [5,10,15],[1,2,3] ])` will return the `list [30, 6]` (because sum of `row 0` is `30` and sum of `row 1` is `6`)

Problem 4: Write a complete program (30 points)

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

To get full credit, your functions should call each other where appropriate to avoid duplicating work!

Your program should contain the following functions:

- `isLower`:
  - Takes 1 input parameter, a string.
  - Returns `True` if the string is all lowercase and `False` otherwise.
  - If the string is empty, function should return `True`.

  Examples of calling this function: `isLower("olympus has fallen")` returns `True` because all characters are lowercase; `isLower("Olympus has fallen")` returns `False` because one character is uppercase (the letter `O`). Note: Non-alphabetic characters should be ignored. For example calling `isLower("olympus has fallen 3 times!!")` still returns `True` because numbers and exclamation marks can be ignored.

- `getMovieName`:
  - Does not have any input parameter.
  - Repeatedly prompts the user for a movie's name until the user enters a string that is not all lowercase
  - Calls `isLower` to check each of the user's entries
  - Returns the movie name

  Each line of user input should count as a single name, even if it has multiple words. For example, the following sequence of user input (in italics and underlined):
  `Enter a movie name:` *mission impossible 2*
  `Enter a movie name:` *batman begins*
  `Enter a movie name:` *The force Awakens*
  should return the string `The force Awakens`

- `main`:
  - Calls `getMovieName` to get a movie name from the user.
  - Prints each word in the movie name that is not all lowercase, in the same line.

  For example, for the movie name `The force Awakens`, the output of the `main` function will be `The Awakens`

Problem 4 code

Problem 4, continued