

Name: (as it would appear on official course roster)	
UCSB email address:	@ucsb.edu
Lab Section:	
Optional: name you wish to be called if different from above	
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")	

h05: Loops, Strings, and Related Exercises

Assigned: Tuesday, May 7th, 2019

Due: Tuesday, May 14th, 2019

Points: 100

- You may collaborate on this homework with AT MOST one person, an optional "homework buddy". MAY ONLY BE TURNED IN THE LECTURE LISTED ABOVE AS THE DUE DATE. There is NO MAKEUP for missed assignments; in place of that, we drop the single lowest score (if you a zero, that is the lowest score.)
- **IMPORTANT:** When submitting this homework:
 - DO NOT USE STAPLES
 - WRITE YOUR **NAME ON EACH PAGE** IN THE SPACE PROVIDED
 - USE DARK INK PENS – PLEASE **DO NOT USE PENCIL**
 - PRINT THIS HOMEWORK **DOUBLE-SIDED** PLEASE!

- | |
|---|
| <ul style="list-style-type: none"> • REMEMBER: If you use code/techniques we have not learned in class, you will NOT get credit! |
|---|

READING ASSIGNMENT: Read Chapter 5, Chapter 4.1, and Chapter 4.2 in Perkovic, review your lecture slides/notes. Then complete these problems.

1. (20 pts) Write a function, **drawTriange()**, that prints a triangle with given height using the character * (WE'RE NOT USING Turtle – instead refer to the example given in class with the rectangle in **lecture 9**). Hint: use 2 nested for loops! Watch your indents! Test your code out! For example **drawTriange(5)** should print:

```

*
**
***
****
*****

```

Name:

(as it would appear on official course roster)
--

2. (20 pts) Use the `countWords(sentence)` function that we talked about in class (*see lecture 10*) and modify it (call it `countWords2(sentence, ln)` – I’ve started it below for you) so that the function takes in an **input string** and an **input integer** as arguments and returns the number of words in the string `sentence` *with length greater than* `ln`. Watch your indents and test this out to be sure it works!

```
def countWords2(sentence, ln):
```

```
    return count
```

Name:

(as it would appear on official course roster)
--

3. (15 pts) Write a Python function, called **NoSpaces()**, that takes in a string as a parameter and returns the string re-written without any space characters. For example:

NoSpaces("I still haven't found what I'm looking for")

returns the string: **"Istillhaven'tfoundwhatI'mlookingfor"**

Requirement: You cannot use the **.replace()** function

Hint: use a for loop to go through each character in the sentence. Watch your indents!

4. (15 pts) Write a more generalizable Python function based on **NoSpaces()** from the previous question, called **NoChar()**, that takes in 2 parameters: a string and a single character (which, technically, is a string too). It returns the first string without any of the characters in it. For example: **NoChar("I'd rather be a hammer than a nail", "a")** returns the string: **"I'd rther be hmmaer thn nil"**

Name:

(as it would appear on official course roster)

5. (10 pts) Write Python statements that print the formatted outputs in (a) thru (e) below, using the already assigned variables **first**, **middle**, and **last**.

```
>>> first = "Martin"  
>>> last = "Prince"  
>>> middle = "Anthony"
```

(a) Prince, Martin Anthony

(b) Prince, Martin A.

(c) Martin A. Prince

(d) M. A. Prince

(e) Prince, M.

Name:

(as it would appear on official course roster)

6. (20 pts) Computational Math Challenge: The number 2^{15} is 32768 and the sum of its digits is $3 + 2 + 7 + 6 + 8$ is 26. Write a program that will ask the user for an integer number between 1 and 1000 (inclusive) and then calculates the sum of the digits of **2 raised to the power** of that number. That is, if a user enters N, the program prints out 2^N and the sum of all the digits in 2^N . For example (user inputs shown in bold):

```
Enter an integer between 1 and 1000: 1000
```

```
The number 2 raised to 1000 is:
```

```
1071508607186267320948425049060001810561404811705533607443750388370351051  
1249361224931983788156958581275946729175531468251871452856923140435984577  
5746985748039345677748242309854210746050623711418779541821530464749835819  
4126739876755916554394607706291457119647768654216766042983165262438683720  
5668069376
```

```
The sum of the numbers in 2 to the power 1000 is: 1366
```

Hint: This is not as difficult as it appears. You know how to calculate 2 to the power N. You may want to convert that calculation into a string, then use that to calculate the sum of all the digits!

