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

h05: Loops, Strings, and Related Exercises

Assigned: Mon. 2/11/19, 9:30 AM
Due: Wed. 2/20/19, 9:30 AM in class
Points: 100

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

- 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.
- When submitting this homework:
  - **DO NOT USE STAPLES**
  - WRITE YOUR **NAME ON <u>EACH</u> PAGE** IN THE SPACE PROVIDED
  - USE **DARK INK PENS** – PLEASE **DO NOT USE PENCIL**
  - PRINT ON **BOTH SIDES** OF THE PAGE!

1. (30 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***). <u>Hint</u>: use 2 nested for loops! Watch your indents! Test your code out! For example **`drawTriangle(5)`** should print:

```
*
**
***
****
*****
```

Name:

(as it would appear on official course roster)

2. (30 pts) Use the **`countWords(sentence)`** function that we talked about in class (***see lectures 9 and 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. (10 pts) Section 5.3 describes how a "list of lists" can be used to represent a 2-dimensional list, such as a matrix or grid.

   Suppose we wanted to program a Python program to play Tic-Tac-Toe. Tic-Tac-Toe is played in a grid similar to the ones shown in the figures below. Figure (1) shows an empty Tic-Tac-Toe grid, while Figures (2 thru 5) show what might be the first three moves of the game. Players take turns filling in squares with 'x' and 'o'.

   We can represent the Tic-Tac-Toe boards shown here as ***lists of lists of strings***. A list of list of strings representation is shown below for three of the five boards. The top level list represents a list of three rows. Each of the lists has three elements, each of which is a space, **x** or **o** for that square.

   The tables below shows the representations for the boards in Figures (1),(3) and (5). Fill in the Python representations for Figures (2) and (4).

<table>
<tr><td><b>Figure 1</b></td><td> | |
-+-+-
 | |
-+-+-
 | |</td><td><code>board = [[' ',' ',' '],[' ',' ',' '],[' ',' ',' ']]</code></td></tr>
<tr><td><b>Figure 2</b></td><td> | |
-+-+-
 | |
-+-+-
x| |</td><td>(   )
<code>board =</code></td></tr>
<tr><td><b>Figure 3</b></td><td> | |
-+-+-
 |o|
-+-+-
x| |</td><td><code>board = [[' ',' ',' '],[' ','o',' '],['x',' ',' ']]</code></td></tr>
<tr><td><b>Figure 4</b></td><td> | |
-+-+-
x|o|
-+-+-
x| |</td><td>(   )
<code>board =</code></td></tr>
<tr><td><b>Figure 5</b></td><td>o| |
-+-+-
x|o|
-+-+-
x| |</td><td><code>board = [['o',' ',' '],['x','o',' '],['x',' ',' ']]</code></td></tr>
</table>

Name:

(as it would appear on official course roster)

4. (10 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”`**
   <u>Hint</u>: use a for loop to go through each character in the sentence. Watch your indents!

5. (10 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  hmmer thn  nil”`**

Name:

(as it would appear on official course roster)

6. (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.
```