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"		

h08: ASCII Characters and Recursion

Assigned: Wed. 3/6/19, 9:30 AM
Due: Wed. 3/13/19, 9:30 AM in class
Points: 100

READING ASSIGNMENT: Read 6.3 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:
   o **DO NOT USE STAPLES**
   o WRITE YOUR **NAME ON __EACH__ PAGE** IN THE SPACE PROVIDED
   o USE **DARK INK PENS** – PLEASE **DO NOT USE PENCIL**
   o PRINT ON **BOTH SIDES** OF THE PAGE!

1. (50 pts) Recall the function **MirrorEncrypt(string)** from class lecture on Monday, 3/6, and that we demonstrated how it works well with lower-case letters, but does not work with upper-case letters. **__Modify it__** so that it can do the same for upper-case letters as it can with lower-case letters. In other words, I want **encrypt("ABCDEFGHIJKLMNOPQRSTUVWXYZ")** to be able to return **"ZYXWVUTSRQPONMLKJIHGFEDCBA"** and I want **encrypt("Penelope")** to be able to return **"Kvmvolkv"**. Assume that these input strings do not have any other characters in them (i.e. just upper-case or lower-case alphabets – nothing else, so no need to check for those). Use the NEXT page to write out your answer.

Name:

(as it would appear on official course roster)

ANSWER TO Q. 1 HERE:

`MirrorEncrypt(string):`

Name:

(as it would appear on official course roster)

2. (50 pts) Examine this recursive function (suggestion: type it in and try it out) and answer the following questions below:

```
def ispal(MyStr):
    if len(MyStr) < 2:
        return True
    if MyStr[0] != MyStr[-1]:
        return False
    else:
        return ispal(MyStr[1:-1])
```

a. (5 pts) What happens when you call it as: **ispal("treert")**?

b. (5 pts) What happens when you call it as: **ispal("madam I'm adam")**?

c. (5 pts) What happens when you call it as: **ispal("Madamimadam")**?

d. (5 pts) What happens when you call it as: **ispal("3loopypool3")**?

e. (5 pts) What is this function looking for (in one sentence)?

f. (5 pts) Identify the base case(s) of this recursive function.

Name:

(as it would appear on official course roster)

g. (20 pts) Write a new function **Check_ispal(word)** that takes a string input (**word**) and: (a) strips all **non-alphanumeric characters** (that is, all characters that are not lowercase/uppercase letters or numbers), from **word**, then (b) converts all uppercase letters in **word** to lowercase letters and finally, (c) calls **ispal()** (the one defined in the previous page) with the modified word string. So, when called this way:
Check_ispal("A man, a plan, a canal: Panama!!"), it returns **True**, and
Check_ispal("A Santa at NASA"), also returns **True**, and
Check_ispal("Yawn... Madonna fan? No damn way!!"), also returns **True**.

I'm giving you a head-start with a "skeleton code" to complete:

```
def Check_ispal(word):
    print(word, end = ": ")
    newword = ''
    for c in word:
```