Name: (as it would appear on official course roster)	Shravan Sharath Shenoy
UCSB email address:	shravansharathshenoy **@ucsb.edu**
Lab Section:	Mon 9:00 AM
Optional: name you wish to be called if different from above	—
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")	—

h08: ASCII Characters and Recursion

Assigned: Thursday, May 30th, 2019
Due: Thursday, June 6th, 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!
- **REMEMBER**: If you use code/techniques we have not learned in class, you will **NOT** get credit!

READING ASSIGNMENT: Read Chapter 6.3 in Perkovic, review your lecture slides/notes. Then complete these problems.

1. (50 pts) Recall the function `MirrorEncrypt(string)` from class lecture 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: Shravan Sharath Shenoy
(as it would appear on official course roster)

ANSWER TO Q. 1 HERE:

```
MirrorEncrypt(string):
    result = ''
    for c in string:
        if ord(c) > 96:
            oc = ord(c)
            or = ord("a") + ord("z") - oc
            result = chr(or) + result
        else:
            oc = ord(c)
            or = ord("A") + ord("Z") - oc
            result = chr(oc) + result
    return result
```

Name: Shravan Sharath Shenoy
(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")`?

True

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

False

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

True

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

False

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

Whether or not the string is mirrored or reversable.

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

-> if len(MyStr) < 2:
-> if MyStr[0] != MyStr[-1]:

Name: Shravan Sharath Shenoy
(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:
        if ord(c) > 47 and ord(c) < 58:
            newword = newword + c
        elif ord(c) > 64 and ord(c) < 91:
            newword = newword + c
        elif ord(c) > 96 and ord(c) < 123:
            newword = newword + c
    newword = newword.lower()
    print(ispal(newword))
```