# Lecture 13 - Exercises

# 13A - Strings and Tuples are Sequences

1. Consider the following function:

def pr(a):
 for c in a:
 print(c, c, end=" ")

The result of calling this function with a particular string a is c c s s c c E E Y Y E E. What was a ?

2. What does the following program print?

```
def fl(tu):
    r = ""
    for v in tu:
        r += str(v + v)
    return r
print(fl(("A", "C", 3)))
```

### 13B - Indexing and Slicing

- 3. Suppose s = "Winter is coming." Which of the following evaluates to "r"?
  - **1**. s[5]
  - 2. s(5)
  - 3. s[6]
  - 4. s(6)
- 4. Suppose <code>last\_name = "Wehrwein"</code> . Which of the following evaluates to <code>"in"</code> ? List all that apply.
  - last\_name[7:8]
  - 2. last\_name[-2:8]
  - 3. last\_name[6:-1]
  - 4. last\_name[-3:]
  - 5. last\_name[6:]
- 5. Which of the following evaluates to the last character of a string s?
  - 1. s[len(s) 1]
  - 2. s[len(s)]
  - 3. s[len(s) + 1]
  - **4**. s[42]
- 6. Suppose last\_name = "wehrwein". For which of the following a and b will last\_name[a] == last\_name[b] evaluate to True ? List all that apply.

```
    a = 1, b = 5
    a = 1, b = 7
    a = 8, b = -4
```

- 4. a = -2, b = -6
- 7. What does the following code print?

```
m = (4, 1, 2, 2, 2)
s = "Scott"
for i in range(len(s)):
    print(s[i] * m[i], end="")
```

8. Consider the following function:

```
def flop(value, number):
    output = ""
    for i in range(number, 0, -1):
        output = output + value[i-1]
    for i in range(number, len(value)):
        output = output + value[i]
    return output
```

Suppose a contains an integer. Which of the following are possible return values of the following call:

flop("no time", a)

#### List all that apply:

- 1. mit one
- 2. nomite
- 3. t onime
- 4. on time
- 5. emit on
- 6. timeno time

## **Problems**

1. Implement the following function:

```
def remove_vowels(string):
    """ Print string, but with all vowels removed. Don't count y as a vowel. """
```

2. Implement the following function, which differs only in that it returns the string without vowels:

```
def remove_vowels(string):
    """ Return string, but with all vowels removed. Don't count y as a vowel. """
```

3. Implement the following function:

```
def count_vowels(string):
    """ Return the number of vowels (not including y) in string. """
```

4. Implement the following function:

```
def remove_comments(string):
    """ Return a copy of string, but with all characters starting with and following
    the first instance of '#' removed. If there is no # in the string, return
    the input unchanged."""
```

5. Write a function that takes a string and prints all prefixes of the string, including the string itself.

```
>>> prefixes("abcd")
a
ab
abc
abc
abcd
>>> prefixes("banana")
b
ba
ban
bana
bana
banan
```

#### 6. Implement the following function:

```
def house_number(address_line):
    """ Return the house number (i.e., digits only) portion of
    the given address line.
    Examples:
        house_number("1600 Pennsylvania Ave")
        => 1600
        house_number("221B Baker St")
        => 221
    """
    # ????
    return result
```

7. Write an <u>add\_banner</u> function that takes a string argument and prints that string, surrounded by a rectangle of *#* symbols. If the string is longer than 80 characters, break the string into multiple lines. You can assume the string does not contain any newlines. An example might look something like the following: