Hands-On Nineteen
Manipulating DNA Sequences

1) Read, understand, and run read_dna.py.
   The program uses built-in python functions: raw_input() and len().
   The raw_input() function reads one line from standard input and returns it as a string (removing the trailing newline).
   - Example: my_dna = raw_input("What is your dna sequence? ")

2) Read, understand, and run complement_1.py.
   The program uses a function: complement that takes one argument: s, of type string.
   The program also uses the dictionary data structure, the for loop, and string concatenation with the "+" operator.

   A **dictionary** consists of `(key, value)` pairs. Dictionaries are delimited by `{` and `}`.
   - Example: basecomp = {'A': 'T', 'C': 'G', 'G': 'C', 'T': 'A'}.

   Elements are retrieved from dictionaries with square brackets `[key]`.
   - Example: basecomp['G']   G is the `key` and C is the `value`

   Function complement uses a **docstring**: the first line in the function, enclosed between a pair of """, that explains the purpose of the function.
   - Example: """Return the complementary sequence string of s."""

3) Read, understand, and run complement_2.py.
   The program uses the dictionary data structure, the for loop, list comprehension and the built-in python function join().

   List comprehension provides a concise way to create lists. Here is an example:
   The following code:

   - Example:

```
squares = []
for x in range(10):
    squares.append(x**2)
```

   is equivalent to:

```
squares = [x**2 for x in range(10)]
```

[https://docs.python.org/2/tutorial/datastructures.html]

4) Read, understand, and run reverse_seq.py.
   The program uses the python built-in functions list(), reverse(), and join().
   The python built-in function:
   - list(): converts a string to a list
   - reverse(): reverses elements of a list.
   - join(): converts a list to a string with option of insertion between characters of string.

5) Read and understand draft_reverse_complement.py. The program does not run. All instances of "XXXXXXXXXX" need to be replaced for the program to run.
The purpose of the program is to construct the reverse complement of a given DNA sequence: CCGGAAGAGCTTACTTAG.
Replace "XXXXXXXXXX" by appropriate code so as to get the program running properly.
Rename the program: reverse_complement.py.

6) Modify reverse_complement.py of problem 5 so as to read the input sequence from the keyboard. In other words, the program should prompt the user to enter his/her sequence from the keyboard rather than having the program find the reverse complement of the hard-coded DNA sequence: CCGGAAGAGCTTACTTAG.
Rename the program: myDNA_reverse_complement.py.