

# Hands-On Nineteen - Programs

## Manipulating DNA Sequences

```
# 1) read_dna.py
# Author: Sami Khuri
# Last updated: January 14, 2016
# Purpose: To output (or print) a DNA sequence read from keyboard
# Program uses built-in python functions raw_input() and len()
my_dna = raw_input("What is your dna sequence? ")

if len(my_dna)>0:
    print "DNA sequence entered is: ", my_dna
else:
    print "You did not enter anything"

#####

# 2) complement_1.py
# Author: Sami Khuri
# Last updated: January 15, 2016
# Purpose: A function to construct the complement of a given DNA sequence
# Program uses a function, a dictionary, for loop, string concatenation
# and a docstring: first line of function enclosed in """

def complement(s):
    """Return the complementary sequence string of s."""
    basecomp = {'A': 'T', 'C': 'G', 'G': 'C', 'T': 'A'}

    seqcomp = ""
    for base in s:
        seqcomp = seqcomp + basecomp[base]
    return seqcomp

dna_seq = 'CCGGAAGAGCTTACTTAG'
print "DNA sequence:\t", dna_seq
print "Its complement:\t", complement(dna_seq)

#####
```





```
# 5) draft_reverse_complement.py
# Author: Sami Khuri
# Last updated: February 13, 2016
# Purpose: To construct the reverse complement of a given DNA sequence
# Program uses dictionary and
#     built-in python functions: list(), join() and reverse(), and
#     user-defined complement(), reverse_seq(), and reverse_comp()
```

```
def complement(s):
    """Return the complementary sequence string of s."""
    basecomplement = {'A': 'T', 'C': 'G', 'G': 'C', 'T': 'A'}
    letters = XXXXXXXXXXXX # Convert s into a list

    letters = [basecomplement[XXXXXXXXXXXX] for base in letters]
    return ".join(letters) # Convert letters into a sequence of characters

def reverse_seq(s):
    """Return the sequence string in reverse order."""
    letters = XXXXXXXXXXXX # Convert s into a list
    letters.reverse()
    return ".join(XXXXXXXXXXXX) # Convert list into a sequence of characters

def reverse_comp(s):
    """Return the reverse complement of DNA sequence s."""
    comp = XXXXXXXXXXXX # Find the complement of s
    revcomp = XXXXXXXXXXXX(comp) # Now reverse the complement of s
    return revcomp
```

```
dna_seq = 'CCGGAAGAGCTTACTTAG'
print "Given DNA sequence:\t", dna_seq
print "Its reverse complement:\t", reverse_comp(dna_seq)
```

#####

- 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.