

Write a series of commands that transform the rna sequence "UUgGAagaGcuuACUUag" to DNA and then calculate its GC content.

```
rna = "UUgGAagaGcuuACUUag"
rna = rna.lower()
rna = rna.replace("u", "t")
float(rna.count("c")+rna.count("g"))/len(rna)
```

Write a set of commands to print the reverse (not the reverse complement) of the sequence 'ACTCGAACGTGTGTCGTTCGGGATTACG'.

```
seq = "ACTCGAACGTGTGTCGTTCGGGATTACG"
seqlist = list(seq)
seqlist.reverse()
reverseseq = "".join(seqlist)
```

This program was supposed to count to ten (it doesn't—it counts to 11).

```
n=0
while n<= 10:
    n=n+1
    print n
```

Fix the above program so that it only counts to 10. There are two ways to accomplish this.

```
n=1
while n <= 10:
    print n
    n += 1
```

```
n=0
while n < 10:
    n=n+1
    print n
```

Let's add up all numbers from 1 to 100 (famous problem).

```
n=0
i=1
while i<= 100:
    n=n+i
    i=i+1
```

Rewrite the while loop that sums all numbers from 1 to 100, as a for loop.
Remember that the range function will give you numbers from 0 to the desired range, not including the endpoint (so range(2) is [1,0]).

```
n = 0
for i in range(101):
    n = n+i
```

Write a set of commands that print the reverse complement of the sequence
'ACTCGAACGTGTGTCGTTCTGGGATTACG'

```
seq = "ACTCGAACGTGTGTCGTTCTGGGATTACG"
seqlist = list(seq)
revcomp = {"A":"T", "T":"A", "C":"G", "G":"C"}
seqlist.reverse()
revcompdna = ""
for i in seqlist:
    revcompdna += revcomp[i]
```

OR you can write a while loop:

```
i=0
while i < len(seqlist):
    revcompdna += revcomp[seqlist[i]]
    i=i+1
```

Write a set of commands that removes all non-acgt letters from the string, "ccF \$334 GGtt".

```
mystring = "ccF $334 GGtt"
dnaletters = "actg"
mystring = mystring.lower()
cleanedstring = ""
for i in mystring:
    if i in dnaletters:
        cleanedstring += i
```

Write a function that cleans a string as above, and returns the reverse complement of the cleaned DNA string.

```
def clean_and_reverse(sequence):
    dnaletters = "ACTG"
    revcomp = {"A":"T", "T":"A", "C":"G", "G":"C"}
    outputstring = ""
    sequence = sequence.upper()
    seqlist = list(sequence)
    seqlist.reverse()
    for i in seqlist:
        if i in dnaletters:
            outputstring += revcomp[i]
    return outputstring
```

```
clean_and_reverse("AC@$5      3TCCC")
'GGGAGT'
```

Extra task:

Add a counter, so that the function reports the length of the longest run of consecutive non-DNA characters.

```
def clean_and_reverse_withcounter(sequence):
    dnaletters = "ACTG"
    revcomp = {"A":"T", "T":"A", "C":"G", "G":"C"}
    outputstring = ""
    sequence = sequence.upper()
    seqlist = list(sequence)
    seqlist.reverse()
    # set the counter that will hold the length
    # of the longest nonDNA string
    counter = 0
    maxcount = 0
    for i in seqlist:
        #check whether the character is in dnaletters. If so, add it
        #to the returned string and reset the counter to zero
        #after saving the value in the counter (if it's longer than
        #the longest nonDNA string seen so far)
        #if the character is not in dnaletters, then
        #increment the counter
        if i in dnaletters:
            outputstring += revcomp[i]
            if counter > maxcount:
                maxcount = counter
```

```
        counter = 0
    else:
        counter += 1
print maxcount
return outputstring
```