Extra practice problems are for beginning programmers who are finding some aspect of the homework challenging. They will break some of the homework problems into smaller pieces and solutions are provided. Try to answer each question yourself before checking against the (provided) answers. There will often be more than one way to accomplish each task. Again, the goal of the homework is to force you to practice thinking through programming problems yourself. Extra practice practice problems are for your own use, and do not need to be turned in.

Problem 1 -
Ask the user for a number. If the number is greater than zero, multiply by 2 and print the result. If the number is less than zero, multiply by -4 and print the result.

Problem 2 -
Ask the user for a 1-letter DNA sequence. Count how many of each nucleotide are in the sequence and print the results.

Problem 3 -
Ask the user to enter a string. Convert all “i”s to “x” and all “o”s to “z”, and print the result.

Try them yourself. Some possible answers are on the next page.

Problem 1 -

```
# Have the user enter a number
number=float(raw_input())

# simple if statement
if number<0 : print number*-4.0
else : print number*2.0
```

================================

Problem 2 -

This problem is a little trickier. There are many different ways to do it. The first part is straightforward:

```
# You haven’t seen this yet, it’s used for the error checking section, not really required
import sys

# read the sequence
seq=raw_input(“Enter sequence: “)

# check the sequence (not really necessary, but good practice)
seq=seq.lower()                # make it all lower case
lets=set(seq)
if not lets.issubset(set((“a”,”g”,”c”,”t”))) :
        print “Error: only AGCT permitted”
        sys.exit(1)

# from here, there are many different approaches. Here is a simple one:
for l in “agct” :
        print l,”: “,seq.count(l)

# here’s a trickier one, which might be better in other circumstances:
counts={}
for l in seq:
        counts[l]=counts.setdefault(l,0)+1

for k in counts.keys():
        print k,”: “,counts[k]
```

================================

Problem 3 -

```
# Again, there are many possible solutions. This is probably the simplest
letters=raw_input()

# The first replace method returns a new string, which is then passed in-turn
# to the second replace method call.
print letters.replace(“i”,”x”).replace(“o”,”z”)
```