Lab 1
Modifying existing programs

Programming isn't just about writing your own programs. Often one of the most useful abilities learning how to program gives you is the capability to modify existing, complex programs to achieve some useful change in behavior.

Below is a program which performs simplistic translation of DNA -> Protein sequence. In this lab we will be extending the capabilities of this program. You can download the program as a text file from the class website. This will be the starting point for this lab.

The program is intentionally not commented, as, unfortunately, many of the programs you run into in the real-world will be sparsely commented. Part of the lab is to figure out how the program works.

```
#!/usr/bin/env python
import sys

xlate={"ttt":"f","ttc":"f","tta":"l","ttg":"l","ctt":"l","ctc":"l","ct
a":"l","ctg":"l","att":"i","atc":"i","ata":"i","atg":"m","gtt":"v","gt
c":"v","gta":"v","gtg":"v","tct":"s","tcc":"s","tca":"s","tcg":"s","cc
t":"p","ccc":"p","cca":"p","ccg":"p","act":"t","acc":"t","aca":"t","ac
g":"t","gct":"a","gcc":"a","gca":"a","gcg":"a","tat":"y","tac":"y","ca
t":"h","cac":"h","caa":"q","cag":"q","aat":"n","aac":"n","aaa":"k","aa
g":"k","gat":"d","gac":"d","gaa":"e","gag":"e","tgt":"c","tgc":"c","tg
g":"w","cgt":"r","cgc":"r","cga":"r","cgg":"r","agt":"s","agc":"s","ag
a":"r","agg":"r","ggt":"g","ggc":"g","gga":"g","ggg":"g"}

fsp=sys.argv[1]
dna=file(fsp,"r").read()
out=file(fsp+".prot","w")

for i in xrange(0,len(dna),3):
      triplet=dna[i:i+3]
      try: amino=xlate[triplet]
      except:
            print "Unknown triplet: ",triplet
            sys.exit(1)
      out.write(amino)

out.write("\n")
```

You should work together as a group to understand how the existing program works, then each person in the group will take one of the tasks listed below and be responsible for the corresponding improvement to the program. One person should be responsible for integrating all of the changes into a single program, then you can test the final program and fix any problems as a group. It is fine (and encouraged) to get advice from others in your group, but each person must write the code for their own task themselves. Make sure everyone understands how the entire modified program works.

Each person should clearly identify the code they wrote with their name in a comment in the final program.

The final modified program for your group should be emailed to the TA with a cc to me just like the homework.

The set of tasks below is in decreasing order of difficulty. If you have 2 people in your group, you must do the first 2 tasks, 3 people, the first 3 tasks, etc. While you are free to decide for yourselves who does which task, you will be turning in a single solution for the entire group, so it would behoove you to have the best programmer tackle the most difficult task.

Tasks in decreasing order of difficulty:

1) Let's assume that we've dealt with identifying a promotor, etc, and that the sequence we're getting is within a few residues of being the start of a coding region of DNA. However, the exact frame hasn't been identified, and clearly if we start with a frame shift we'll get the wrong sequence. Modify the program to start with the correct frame by assuming the first ATG we find represents the beginning of the coding region.
2) Similarly, translation will only occur until a stop codon has been reached (TAA, TAG, TGA). Have the program terminate when it reaches a stop codon (and don't have it print anything for the stop).
3) Very frequently sequence data is formatted with numbers, spaces, carriage returns, etc. Strip out all of the non-DNA characters from the input data before beginning the translation process.
4) The current program assumes all of the input and output letters are in lower-case. Modify the program so it will accept any mix of upper and lower-case characters, and the output is upper-case.

- If your group had 2 people, test your program with:
gatggcagctaaagacgtaaaatgaaaa

- If your group had 3 people, test with:
1   gatggcagct aaagacgtaa aatgaaaa

- If your group had 4 people, test with:
1   gatGGcagCt aAagaCgtaa aAtgaaaa

- In all cases, it should produce:
maakdvk

as output (in upper case if you had 4 people)