

## Python 5 Problem Set

1. In the file `Python_06_nobody.txt` find every occurrence of ‘Nobody’ and print out the position.
2. In the file `Python_06_nobody.txt` substitute every occurrence of ‘Nobody’ with your favorite person’s name and save the file as `YourFavoritePersonName.txt` (ex. `Michael.txt`).
3. Find all the lines in a `Python_06.fasta` that are the header (`>seqName desc`) using pattern matching.
4. If a line matches the format of a FASTA header, extract the sequence name and description using sub patterns.
  - Print id:“extracted seq name” desc:“extracted description”
5. Create or modify your FASTA parser to use regular expressions. Also make sure your parser can deal with a sequence that is split over many lines.
6. The enzyme `ApoI` has a restriction site: `R^AATTY` where R and Y are degenerate nucleotides. See the IUPAC table to identify the nucleotide possibilities for the R and Y. Write a regular expression to find and print all occurrences of the site in the following sequence `Python_06_ApoI.fasta`.

```
seq1          GAATTCAAGTTCTTGTGCGCACACAAATCCAATAAAAAC-
TATTGTGCACACAGACGCGAC          TTCGCGGTCTCGCTTGTTCTTGTTG-
TATTCGTATTTTCATTTCTCGTTCTGTTTCTACTT          AACAATGTGGT-
GATAATATAAAAAATAAAGCAATTCAAAAGTGTATGACTTAATTAATGA      GC-
GATTTTTTTTTTTGAAATCAAATTTTTTGGAACATTTTTTTTTTAAATTCAAATTTTG-
GCGA      AAATTCAATATCGGTTCTACTATCCATAATATAATTCATCAGGAATA-
CATCTTCAAAGGC  AACGGTGACAACAAAATTCAGGCAATTCAGGCAAATAC-
CGAATGACCAGCTTGGTATC      AATTCTAGAATTTGTTTTTTGGTTTTTTATT-
TATCATTTGTAAATAAGACAAACATTTGTTC      CTAGTAAAGAATGTAACACCA-
GAAGTCACGTAAAATGGTGTCCCCATTGTTTAAACGGTT  GTTGGGACCAATG-
GAGTTCGTGGTAACAGTACATCTTTCCCCTTGAATTTGCCATTCAAA ATTTGCG-
GTGGAATACCTAACAAATCCAGTGAATTTAAGAATTGCGATGGGTAATTGACA
TGAATTCCAAGGTCAAATGCTAAGAGATAGTTTAATTTATGTTTGAGACAAT-
CAATTTCC      CAATTTTTCTAAGACTTCAATCAATCTCTTAGAATCCGCCTCTG-
GAGGTGCACTCAGCCG      CACGTCGGGCTCACCAAATATGTTGGGGTTGTGCG-
GTGAACTCGAATAGAAATTATTGTGCG  CCTCCATCTTCATGGCCGTGAAATCG-
GCTCGCTGACGGGCTTCTCGCGCTGGATTTTTTC ACTATTTTTGAATACATCAT-
TAACGCAATATATATATATATATATTTAT
```

7. Determine the site(s) of the physical cut(s) by `ApoI` in the above sequence. Print out the sequence with “^” at the cut site.

Hints: Use `sub()` Use subpatterns (parentheses and `group()` ) to find the cut site within the pattern.

Example: if the pattern is `GACGT^CT` the following sequence

```
AAAAAAAAGACGTCTTTTTTTAAAAAAAAGACGTCTTTTTTTT
```

would be cut like this:

```
AAAAAAAAGACGTCTTTTTTTAAAAAAAAGACGTCTTTTTTTT
```

8. Now that you’ve done your restriction digest, determine the lengths of your fragments and sort them by length (in the same order they would separate on an electrophoresis gel).

Hint: Convert this string:

```
1 AAAAAAAGACGT~CTTTTTTAAAAAAGACGT~CTTTTTT
```

Into this list:

```
1 ["AAAAAAGACGT", "CTTTTTTAAAAAAGACGT", "CTTTTTT"]
```

9. Download this file of enzymes and their cut sites to fill a dictionary of enzymes paired with their recognition patterns. Be aware of the header lines, and be aware of how the columns are delimited.
10. Write a script which takes two command line arguments: the name of an enzyme and a fasta file with a sequence to be cut. If the provided enzyme is present in the hash, and will act successfully on the provided sequence, print out:
  - the provided sequence, annotated with cut sites
  - the number of fragments
  - the fragments in their natural order (unsorted)
  - the fragments in sorted order (largest to smallest)