

CSC241 Winter 2014 - Lab Assignment 8

1 Program 1 a8p1.py (1 points)

Write a version of a palindrome recogniser that accepts a file name from the user, reads each line, and prints the line to the screen if it is a palindrome.

2 Program 2 a8p2.py (1 points)

According to Wikipedia, a semordnilap is a word or phrase that spells a different word or phrase backwards. (“Semordnilap” is itself “palindromes” spelled backwards.) Write a semordnilap recogniser that accepts a file name (pointing to a list of words) from the user and finds and prints all pairs of words that are semordnilaps to the screen. For example, if “stressed” and “desserts” is part of the word list, the the output should include the pair “stressed desserts”. Note, by the way, that each pair by itself forms a palindrome!

3 Program 3 a8p3.py (1 points)

Write a procedure `char_freq_table()` that, when run in a terminal, accepts a file name from the user, builds a frequency listing of the characters contained in the file, and prints a sorted and nicely formatted character frequency table to the screen.

4 Program 4 a8p4.py (1 points)

Write a program that takes two strings as input. The first is a filename. It then open the file, reads its lines and prints the lines that contain the second input string.

5 Program 5 a8p5.py (2 points)

Write a program that takes two strings as input. The first is a filename. It then open the file, reads its lines and prints the lines that contain any of the

characters in the second string.

6 Program 6 a8p6.py (2 points)

Consider the following Python dictionary (also found in file `/tmp/ami.py`):

```
AMI = {'Ala/A': ['GCT', 'GCC', 'GCA', 'GCG'],
       'Leu/L': ['TTA', 'TTG', 'CTT', 'CTC', 'CTA', 'CTG'],
       'Arg/R': ['CGT', 'CGC', 'CGA', 'CGG', 'AGA', 'AGG'],
       'Lys/K': ['AAA', 'AAG'],
       'Asn/N': ['AAT', 'AAC'],
       'Met/M': ['ATG'],
       'Asp/D': ['GAT', 'GAC'],
       'Phe/F': ['TTT', 'TTC'],
       'Cys/C': ['TGT', 'TGC'],
       'Pro/P': ['CCT', 'CCC', 'CCA', 'CCG'],
       'Gln/Q': ['CAA', 'CAG'],
       'Ser/S': ['TCT', 'TCC', 'TCA', 'TCG', 'AGT', 'AGC'],
       'Glu/E': ['GAA', 'GAG'],
       'Thr/T': ['ACT', 'ACC', 'ACA', 'ACG'],
       'Gly/G': ['GGT', 'GGC', 'GGA', 'GGG'],
       'Trp/W': ['TGG'],
       'His/H': ['CAT', 'CAC'],
       'Tyr/Y': ['TAT', 'TAC'],
       'Ile/I': ['ATT', 'ATC', 'ATA'],
       'Val/V': ['GTT', 'GTC', 'GTA', 'GTG']}
```

It maps the name of aminoacids to the corresponding codons (DNA subsequences of size 3). Write a program that reads as input a DNA string, breaks it into codons (substrings of size 3) and counts the corresponding aminoacids. It should output aminoacids (sorted alphabetically) and their frequency (separated by a colon, one per line). Example:

```
input> ATGATGAAA
output>
Lys/K: 1
Met/M: 2
```

You can use the following syntax so loop over the keys of a dictionary `d` in sorted order:

```
for key in d: print(k,d[key])
```