PSG College of Technology

Department of Applied Mathematics and Computational Sciences

Python Programming Laboratory

Problem Sheet

1. Write a function called most frequent that takes a string and prints the letters in decreasing order of frequency
2. Write a recursive Python function that has a parameter representing a list of integers and returns the maximum stored in the list. Thinking recursively, the maximum is either the first value in the list or the maximum of the rest of the list, whichever is larger. If the list only has 1 integer, then its maximum is this single value, naturally.
3. Create a dictionary as given below:

Given a DNA sequence string, calculate the frequency of each base pair (i.e. the number of times that each letter appears in the sequence).

base\_freq("AAGTTAGTCA")

{"A": 4, "C": 1, "G": 2, "T": 3}

1. Create a dictionary as given below:

alonzo = {"age": 10, "height": 42, "weight": 175, "instrument ": "fiddle" }

turing = {"age": 41, "height": 70 "weight": 160, "instrument": "theremin"}

bertha = {"age": 32, "height": 97, "weight": 587, "instrument": "cello"}

tinkerB = {"age":100, "height": 4, "weight": 0.5, "instrument": "cello"}

banditos = {"Alonzo": alonzo, "Turing": turing, "Bertha": bertha, "TinkerB": tinkerB}

Given a dictionary of people, like the one above, write a function that returns a new dictionary that contains only the people who play a certain instrument.

1. It's time for a cringeworthy highschool favorite. Your class decides to rank everyone in the class in order of preference. As class nerd, you've been assigned the task of finding the most popular student. Given a dictionary of each student and a corresponding dictionary of their preferences, write a function that returns a student with the lowest average score. Test this function with function number C3.

>>> alice\_ratings = {"alonzo": 1, "bob": 3, "turing" : 2}

>>> bob\_ratings = {"alice": 1, "alonzo": 2, "turing": 3}

>>> alonzo\_ratings = {"alice": 3, "bob": 2, "turing": 1}

>>> turing\_ratings = {"alice": 2, "alonzo": 1 "bob": 3}

>>> friends = {"alice": alice\_ratings, "bob": bob\_ratings, "alonzo": alonzo\_ratings, "turing": turing\_ratings}

>>> most\_popular(friends)

"alonzo"

1. Two words form a “metathesis pair” if you can transform one into the other by swapping two letters; for example, “converse” and “conserve.” Write a program that finds all of the metathesis pairs in the dictionary.