Make-Up Midterm Preparation

(1) Write a function that calculates the *n*-th harmonic number, defined to be $\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\ldots+\frac{1}{n-1}+\frac{1}{n}$.

(2) Write a function of a single argument, a string, that removes all vowels from the string. For example, given 'Marquette University', the function returns 'Mrqtt nvrst'. Capitalization is not affected by the function.

(3) Write a function of a single argument, a string. The function returns the same string, but with all the first letters raised to capital letters. For example: 'The red fox jumps over the blue cat.' becomes 'The Red Fox Jumps Over The Blue Cat.'

(4) Write a function that functions as a desk calculator. It repeatedly asks the user for a number to input. If the user does not enter a number, it just says so. When the user enters 'stop', then it prints out the sum of the numbers entered. (Hint: Obviously, we need to create an infinite while loop. In the loop, we ask for input, check it, and process it. When the user enters 'stop', we break out of the loop. When the user enters a wrong number, we use a 'continue' command.

(5) Use list comprehension to generate the following lists, sets, and dictionaries:

(1) The set of the first 20 cubes starting with 1^3 and finishing with 20^3.

(2) The set of all letters in a string

(3) The list of all reciprocals `[1, 0.5, 0.333333, 0.25, 0.2, … ]` of numbers 1 to 100 (included).

(4) The list of all numbers between 1 and 100 (included) whose square and cube are both 1 modulo 3.

(5) A dictionary that assigns to all letters the value 0. The letters are in string.ascii_letters. `{'a':0, 'b':0, 'c':0, …  'Y':0, 'Z':0}.`

(6) Write a function `make_dictionary(list_of_keys, list_of_values)` that takes two lists (of equal length) as input and creates a dictionary that associates the first element of the first list to the first element of the second list, etc. For example, `make_dictionary([“a”, “b”, “c”],[1,2,3])` returns `{“a”:1, “b”:2, “c”:3}.`

(7) Write a function that opens a text file, removes anything but letters and newline characters, converts are letters to uppercase, and writes the result into a new file. The function takes the name of an input text file and of an output text file as parameters. `def clean(filename_in, filename_out).`

(8) Write a function of a single argument, a file name. The function opens the corresponding file and prints out all integers in the file. Since we do not want punctuation marks to hide an integer, you need to strip punctuation from each word. You can use `string.punctuation` for this. You need to use an exception to check whether a stripped word can be made into an integer.

(9) Write a function that takes as input a csv file. Each line in the csv file has some numbers, separated by commas. The function returns the average of all the numbers in the file. If the file contains non-numbers, they are ignored.

(10)Write a function that counts all letters in a file. A letter is defined to belong to `string.ascii_letters.`