Day 4

1. Write a function *attached*() that takes three parameters, the first an integer (required parameter), the second a string (required parameter), and the third an optional parameter whose default vale is ":". Returned value will be the first parameter, concatenated with the second, using the third.

2. Give an implementation of a function called *nums*() that has three parameters, the first, an integer, is required, the second parameter *multInt*, an integer, is optional with a default value of 10, the final parameter *switch*, is also optional with a default value of *False*. The function should multiply the two integers together, and if switch is True, should change the sign of the product before returning it.

3. The p-norm of a vector v = (v_1, v_2, ..., v_n) in n-dimensional space is defined as

$$\|v\| = \sqrt[p]{v_1^p + v_2^p + \cdots + v_n^p}.$$

   For the special case of p = 2, this results in the traditional Euclidean norm, which represents the length of the vector. For example, a two-dimensional vector with coordinates (4,3) has a Euclidean norm of $\sqrt{(4^2+3^2)}= 5$. Give an implementation of a function named *norm*() such that norm(v, p) returns the p-norm value of v and norm(v) returns the Euclidean norm of v.

4. Define a function to return a tuple containing a pair, the sum of all the even integers and odd integers respectively, in a list of integers passed as an argument to the function.

5. Write a function that accepts a list and returns a new list with unique elements of that list.

6. Write a function *lastChar*() that takes a string as input, and returns only its last character. Use this function to sort list of strings by the last character of each string, from highest to lowest.

   Sample   Input: ['150', '23', '781', '19', '3478', '12', '9005', '84', '9817', '96']

             Output: ['19', '3478', '9817', '96', '9005', '84', '23', '12', '781','150']

7. Redo assignment #6 with lambda.

8. Sort a list of numbers based on their absolute values, writing your own function for calculating absolute value of a number and (i) using it in sorted() without lambda, (ii) using it in sorted() with lambda.

   Sample   Input: [-5, -7, 4, -2, -9]

                   Output: [-2, 4, -5, -7, -9]

9. Sort a list of words first by their length, smallest to largest, and then alphabetically to break ties

among words of the same length (using sorted() and lambda).

Sample   Input: ['mtech', 'btech', 'mca', 'bca', 'diploma', 'dsc']

                     Output: ['bca', 'dsc', 'mca', 'btech', 'mtech', 'diploma']

10. Sort a list of words first by their length, largest to smallest, and then alphabetically to break ties among words of the same length (using sorted() and lambda).

Sample   Input: ['mtech', 'btech', 'mca', 'bca', 'diploma', 'dsc']

                     Output: ['diploma', 'btech', 'mtech', 'bca', 'dsc', 'mca']

11. Given a dictionary as stated in the sample input,

     (i)   sort the states in order by the first city name. (Sample Output: ['West Bengal', 'Maharastra','Kerala'])

     (ii)   sort the states by the length of the second city name, break ties of equal length by name of the second cities. (Sample Output: ['Maharastra', 'West Bengal', 'Kerala'])

     (iii)   sort the states in order by the number of cities having length greater than 6. (Sample Output: ['Maharastra', 'Kerala', 'West Bengal'])

         Sample   Input: {"Kerala": ["Kannur", "Palakkad", "Thalassery"],

                                  "Maharastra": ["Bhandara", "Nagpur","Wardha"],

                                  "West Bengal": ["Asansol", "Basirhat", "Bardhaman"]}

12. Write a function that takes a string as a parameter and returns a list of the five most frequent characters in the string.

13. Sort a list of roll numbers by the last three digits of the roll number.

Sample   Input: [20223005, 20222342, 20229000, 20220002, 20222345, 20229329]

                Output: [20229000, 20220002, 20223005, 20229329, 20222342, 20222345]

14. Sort a list of names alphabetically by last name.

Sample   Input: ['Ales Bialiatski', 'Alain Aspect', 'Anton Zeilinger', 'Douglas Diamond']

                Output: ['Alain Aspect', 'Ales Bialiatski', 'Douglas Diamond', 'Anton Zeilinger']