

## Data Structure with Python Week 6

1. Write a Python program to sort a tuple by its float element.  
Sample data: [('item1', '12.20'), ('item2', '15.10'), ('item3', '24.5')]  
Expected Output: [('item3', '24.5'), ('item2', '15.10'), ('item1', '12.20')]
2. WAP to Create Set from Tuple.
3. Write a function group(list, size) that take a list and splits into smaller lists of given size.
4. Write a function extsort to sort a list of files based on extension
5. Write a program reverse.py to print lines of a file in reverse order.
6. Provide an implementation for zip function using list comprehensions.
7. Python provides a built-in function filter(f, a) that returns items of the list a for which f(item) returns true. Provide an implementation for filter using list comprehensions.
8. Write a function enumerate that takes a list and returns a list of tuples containing (index,item) for each item in the list.
9. Write a function array to create an 2-dimensional array. The function should take both dimensions as arguments. Value of each element can be initialized to None:
10. Write a function mutate to compute all words generated by a single mutation on a given word.  
A mutation is defined as inserting a character, deleting a character, replacing a character, or swapping 2 consecutive characters in a string. For simplicity consider only letters from a to z
11. Write a function nearly\_equal to test whether two strings are nearly equal. Two strings a and b are nearly equal when a can be generated by a single mutation on b.
12. Write a Python program to count the elements in a list until an element is a tuple.