# PPL Labsheet – 2

1. Insert an element at a given position into a list. If position is negative is negative insert the element in reverse order, that is:
   say (insert-at 'a '( b c d e) -1) outpout will be (b c d e a).
2. Write the Quick sort algorithm in scheme.
3. Define a procedure “shuffle-list” in scheme, which takes two lists as arguments and returns a single list, the elements of which are drawn alternately from the two given lists as long as they both hold out, after which the rest of the elements are the same as those of the longer of the given lists. For example, the value of the call (“shuffle-list” '(x y z) '(a b c d e f g)) is (x a y b z c d e f g).
4. Define a Scheme procedure “merge-list” which takes two lists of real numbers as arguments, each list in ascending order, and returns a single list, which contains all the elements from both lists, in ascending order.
5. Define a Scheme procedure “filter-list”, which takes a predicate and a list as arguments, and returns a list that contains the elements of the given list that satisfy the given predicate (that is, the elements for which applying the predicate to the element would return #t). For example, (“filter-list” odd? '(1 8 6 3 4 11 10)) should return (1 3 11).