# CO1107 Algorithm, Data Structure & Advanced Programming - Workshop Week 5

**Task 1:**
The function **reverse_list** takes a list and produces the same list but with the elements in reverse order (compared to the order they are in the original list). For example, the the list [3,1,2] becomes [2,1,3] . Use a while loop to implement this function.

**Task 2:**
Modify your program at Task 1; Implement the same function, but this time using a stack. Write the scripts to test your function for all sorts of input.

**Task 3:**
The function **minsort** works as follows. Given an input list of elements to sort, it scans the list to find the smallest element. It removes this from the list and adds it to the end of the result list which is eventually returned (the result list is initially empty). It repeats this step until the input list is empty, at which point the result is returned.
Implement this function.

**Task 4:**

In lectures we saw the **mergesort** function which sorts a list of integers in increasing order. Implement the **reverse_mergesort** function so that it sorts the list in decreasing order. Do not just call reverse on the result of mergesort . Instead, think about how mergesort compares elements to see which one is smaller during the **merge** step, and try to change this so that the elements are sorted in decreasing order rather than increasing order.

**Task 5: Test 1 Question**

**Part A)**

For the following program, **show the state of the queue Q after each iteration of the loop**. You may assume the following.

- append (aQueue,A): will append A into the queue aQueue
- serve(aQueue): will serve from the queue aQueue
- size (aQueue): will return the number of items in the queue aQueue

**Part B)**

Re-write the code by replacing those assumed functions listed above with the proper python function.