# Algorithms Assignment

**Q2 –**

In a flowchart I have an algorithm that combines the four lists(array of structures that I made in the previous question) into one main list, I created another array of structures of the size of the maximum amount of students in all modules ”allsurnames”.Then I have sorted those surnames alphabetically, using the merge sort.

a) I used the merge sort to sort the surnames alphabetically, as it is a big amount of data 42 surnames maximum, the merge sort seemed the adequate as it sorts the data quicker, and accurately.

b) The big O of the merge sort is O(nLogn), it is more efficient it repeatedly divides the data by halves and then swaps the data in the correct order. It is the fastest sort algorithm in comparison to other studied in the module (bubble sort, insertion sort),it also works wwll in bigger data sizes such as this one.
The algorithm divides the list in two (left- right), then it assigns the index number for the low, middle and high position in the list. The big O, analyses the efficiency of the algorithm “n” is the variable that suggest the number of things.
The merge sort is in reality a combination of two algorithms.(O(n) x O(logn))
This sorting algorithms big O is is O(nLogn), because the divide and conquer part of the sort is O(log n) and the combination part is O(n) .