**CSCI 301 Moustafa Elsayed  
Computer Science 2**

**Merging two sorted sequences**

**Introduction**

Merging two sorted sequences is taking each sequence and adding them to a third sorted sequence. This program reads two input files, that the user chooses, then adds them to an initial array. Then, the array is sorted ascendingly in order to start moving the integers in the initial array to the new and final array. During this process, the program compares an integer from the initial array with the integers in the final array, if the program found a similar integer in the final array it moves to the next number, if no similar integer was found then the program adds this integer to the final array and then it moves on. Finally, after successfully moving all the integers to the final array, the program shows the output on the compiler then writes the final array to an output folder.

**Data Structures**

The program uses two data structures, an array to hold all the integers from the two input files and an array to hold the integers from the first array but without repeating any integer. There are five variables, the *temp* variable is to temporary hold the integer from its input file in order to add it to the initial array. A *counter* variable to keep track of the number of integers in both sequences. A *distinctcount* to keep track of all the non-repeated integers. A *result* Boolean in order to signal the program if there is a repeated integer. A *filename* variable to hold the filenames, this variable gets re-written with the second input file name after the first input file name is assigned.

**Functions**

This program uses three functions. The *searchSim()* function, compares every integer from the initial array with the integers from the final array, if it finds a similar integer then the function returns true and if it could not find a similar integer the function returns false. The *sort()* function, sorts the array in ascending order by holding each integer from the array and comparing it with the other integers in the array, if the function finds a smaller integer it swaps them. The *swap()* function holds the value of the first integer in a temporary variable, then assigns the value of the second integer to the first integer, then the temporary variable assigns its value to the second integer.

**The main program**

The main program asks the user for the name of the first input file then opens that file, then asks for the name of the second input file and opens that file. After that, the program opens the output file. The program opens a loop that only ends after all the data in the first input file has been gone through, assigns an integer to a temporary variable and that variable assigns the integer to the initial array with the counter variable’s value indicated which space in the array it is assigned to, then the counter increments. The program opens the same loop but with the other input file as its parameter. The program calls the sorting function to make sure all the integers in the initial array are in ascending order. The program opens another loop with the counter as its parameter, inside this loop the search function is called then returns a Boolean value to a variable, an if function is called to assign the value of the initial array to the final array and increments the distinct counter only if the Boolean variable is false. If the Boolean variable is true the loop moves to the next integer. A final loop is called in order to output the results to the user and write the final array in the output file and a message informing the user how many distinct numbers are there in these two sequences.