

# Day 8 - Programs at Bootcamp

## Section C - String and Arrays :- Sorting and Search Functions

1. Write a static function to Binary Search the Word from Word List
  - a. Desc -> Read in a list of words from File. Then prompt the user to enter a word to search the list. The program reports if the search word is found in the list.
  - b. I/P -> read in the list words comma separated from a File and then enter the word to be searched
  - c. Logic -> Use Arrays to sort the word list and then do the binary search
  - d. O/P -> Print the result if the word is found or not
2. Write a static function to Bubble Sort. Reads in integers prints them in sorted order using Bubble Sort and output the result
3. Write a program with Static Functions to do Merge Sort of list of Strings.
  - a. Logic -> To Merge Sort an array, we divide it into two halves, sort the two halves independently, and then merge the results to sort the full array. To sort  $a[lo, hi)$ , we use the following recursive strategy:
  - b. Base case: If the subarray length is 0 or 1, it is already sorted.
  - c. Reduction step: Otherwise, compute  $mid = lo + (hi - lo) / 2$ , recursively sort the two subarrays  $a[lo, mid)$  and  $a[mid, hi)$ , and merge them to produce a sorted result.

# Day 8 - Programs at Home

## Section C - String and Arrays :- Sorting and Search Functions

1. Write a static function to do Binary Search of Integers and then Question to find your number (Magic Number)
  - a. Desc -> takes a command-line argument N, asks you to think of a number between 0 and N-1, where  $N = 2^n$ , and always guesses the answer with n questions.
  - b. I/P -> the Number N and then recursively ask true/false (Is Less Y/N? ) if the number is between a high and low value
  - c. Logic -> Use Binary Search to find the number
  - d. O/P -> Print the intermediary number and the final answer
2. Write a static function to do Insertion Sort
  - a. Desc -> Reads in strings from standard input and prints them in sorted order. Uses insertion sort.
3. Write FrequencyCount.java program that reads a sequence of strings from standard input and then prints a table of the distinct values found and the number of times each was found, in decreasing order of the frequencies. We accomplish this by two sorts.

**Computing the frequencies** - Our first step is to sort the strings on standard input. In this case, we are not so much interested in the fact that the strings are put into sorted order, but in the fact that sorting brings equal strings together. If the input is

To Be Or Not To Be

then the result of the sort is

Be Be Not Or To To

Now, with equal strings all together in the array, we can make a single pass through the array to compute all the frequencies. Store the results in an Array of Pair class which has a Key and a Value.