a. A brief description of notable obstacles you overcame.

At first, I was having difficulty in finishing the matchingValuesTogether function, and I came up with the idea that: basically, I need to move the index variable and determine if the element right next to it is an identical one or not. (accomplished by an if-else statement) Then, if identical, I will simply push the index one step forward and repeat this process, otherwise, I will detect if the rest element is identical and change the return result accordingly. Thus, by changing the problem to looking for discrete identical elements, I sort of solved this problem.

Secondly, I found it some difficult to solve the divide function. After reading the assignment and FAQ and discussions, I was inspired to sort the array first by defining an additional function to find the largest string in the array. Initially, I was trying to find the minimum, but then I found it work the index from the array[0] so I changed the function to find the max string, then finally changed to find the index of the largest string. In this way, I could sort from the end of the array and sort it from largest to smallest by swapping the value or string within the array as I push the index from n-1 to zero and changing the size of array that I passed to the max\_index finder function. Then the divide functionality is just to find the first index as the string there is equal or greater than divide.

b. A list of the test data that could be used to thoroughly test your functions, along with the reason for each test.

1. string people[5] = { "a" }; testing for array with a single string; testing n<=0 cases.
2. string people[5] = { "samwell", "jon", "margaery", "daenerys", "tyrion" }; testing with assignment example array and all functions with given input.
3. string folks[8] = { "samwell", "jon", "margaery", "daenerys", "tyrion", "sansa", "howard123", "jon" }; testing with assignment example array and all functions with given input.
4. string people[5] = { "" }; testing empty array.
5. string people[5] = { "a","a","b","c","d" }; testing for matchingValueTogether function and duplicate function.
6. string people[5] = { "5", "4", "3", "2", "15" }; testing for decreasing and divide function with (people,6,”21”).
7. string people[5] = { "c", "b", "b", "a","a" }; testing for decreasing function.
8. string people[6] = { "gamma", "tau", "alpha", "beta", "beta", "eepsilo" }; testing divide with (people,6,”foo”) and other functions.
9. string people[6] = { "gamma", "tau", "", "beta", "beta", "eepsilo" }; testing if an empty string would cause any bug.
10. string people[2] = { "sss", "s" }; testing countSs.
11. string people[5] = { "a","a","b","a","d" }; testing for matchingValueTogether if identical strings are separated.
12. string people[5] = { "20", "2", "3", "4", "35"}; testing for shiftLeft with 2, "X" as input.