a. A few of the obstacles I overcame while writing and testing this code was the conditions for the loops and which loops to use in which instance. For example, during the makeMerger function I was stuck for hours looking at a for loop I had made and kept getting an incorrect output, but I couldn’t figure out why. I spent way too much time trying to make the for loop work, and when I eventually gave up and used an if statement instead, the code ran perfectly. In summation I overcame many obstacles during this project, but the largest obstacle was my own stubbornness. Countless times, I was stuck staring at the code I had written, trying the same thing over and over getting the same result. It wasn’t until I swallowed my pride and took it in a different direction, that I realized how simple my errors were.

b. Test Cases

enumerate:

string a[5] = {“neil”, “sonia”, “john”, “amy”, “”, “elena”, “john”};

(a, 5, “sonia”) normal test case

(a, 3, “neil”) integer is smaller than the size of array

(a, 1, “”) empty string and smaller than array size

(a, 5, “”) just empty string

(a, 2, “john”)

findMatch:

(a, 5, “elena”) simple test

(a, 2, “sonia”) length is less than array length

(a, 3, “john”) word is in array multiple times

(a, 0, “amy”) array length is zero

(a, 5, “chris”) word is not in array

findRun:  
(a, 5, “elena”, int b, int a)

(a, 3, “”, int b, int a) empty string test

(a, 5, “john”, int x, int y) no runs found

findMin:

(a, 4)

(a,5)

(a,0) length is zero

(a,2)

(a,1)

moveToEnd:

(a, 5, 3)

(a, 5, 5) position is out of bounds

(a, 0, 0) both zeros

(a, 0, 3)length is zero but position is not

moveToBeginning:  
(a, 5, 3) normal test

(a, 5, 5) position is out of bounds

(a, 0, 0) both are zero

(a, 0, 3)position is larger than length

removeDups:  
(a,5)

(b, 7) array with no duplicates

(b, 0) array with length 0

(b, 7) array with duplicates but not in a row

(b,7) array with multiple sets of duplicates in a row

subsequence:

(a, 6, b, 3) normal test

(a, 4, b, 6) second array is larger than first

(a, 0, b, 4) first array is length 0

(a, 4, b, 0) second array is length 0

makeMerger:

(a, 6, b, 3, c, 10) normal test

(a, 6, b, 3, c, 2) max is not large enough

(a, 0, b, 3, c, 10) one array is zero

(a, 0, b, 0, c, 10) both arrays are zero

divide:

(a, 5, “g”) normal test

(a, 3, “f”) length is shorter than array

(a, 0, “g”) length is zero

(a, 5, “g”) e.g. array a has the letter g as one of its strings- tests when an index is the exact same as the divider

(a, 1, “c”) e.g. array a has one element and it is the letter b- tests when there is not an index greater than the divider and if the function returns n