1. There were a lot of obstacles throughout the building of this project. One of the obstacles in this project that came up in a lot of the methods was when I had to change the value in two of the elements of the array. But when you put the information in one of the elements in the other, the information in the second array is lost. I solved this problem by creating a temporary variable which stores the information that is lost in the temporary variable. Also, I had some trouble with returning the smallest index, that has some particular quality. For this, I had to use the break command for many of the methods, as soon as some condition is met.
2. string h[7] = { "selina", "reed", "diana", "tony", "", "logan", "peter" };

assert(lookup(h, 7, "logan") == 5);

assert(lookup(h, 7, "diana") == 2);

assert(lookup(h, 2, "diana") == -1);

string r[3] = { "selina", "reed", "selina" };

assert(lookup(r, 3, "selina") == 0);

assert(positionOfMax(h, 7) == 3);

string g[4] = { "selina", "reed", "peter", "sue" };

assert(differ(h, 4, g, 4) == 2);

assert(appendToAll(g, 4, "?") == 4 && g[0] == "selina?" && g[3] == "sue?");

assert(rotateLeft(g, 4, 1) == 1 && g[1] == "peter?" && g[3] == "reed?");

string e[4] = { "diana", "tony", "", "logan" };

assert(subsequence(h, 7, e, 4) == 2);

assert(subsequence(r, 3, g, 4) == -1);

assert(subsequence(g, 4, r, 3) == -1);

string d[5] = { "reed", "reed", "reed", "tony", "tony" };

assert(countRuns(d, 5) == 2);

string f[3] = { "peter", "diana", "steve" };

assert(lookupAny(h, 7, f, 3) == 2);

assert(lookupAny(f, 3, h, 7) == 0);

assert(flip(f, 3) == 3 && f[0] == "steve" && f[2] == "peter");

assert(split(h, 7, "peter") == 3);

assert(split(g, 4, "reem") == 2);

assert(split(g, 0, "peter") == 0);

assert(split(g, 4, "zed") == 4);

Test Cases that would not work is anything where the number of elements put in as a parameter is greater than the number of elements in the array.