Eduardo Ramirez Torres

UID: 205317677

CS31

**Report for Project 4**

1. Project 4 wasn’t as bad as Project 3. It was relatively simple to understand. I did, however, run into problems along the way. The flip function took a bit to code as I had to mess around with organizing the array in such a way that produced desirable results. I ran a for loop that swapped the last and first positions, with each iteration getting closer to the center. Figuring out what to do at the center was what took quite a while to think through, but I eventually figured it out. In addition to this, I tried doing the extra credit. While I was able to do it for flip and rotate left, I just couldn’t figure out how to do it for separate. I wracked through my mind to see if there was a way to do it, but I decided to leave it as is. Overall, Project 4 was a success.
2. Test Cases:
   1. Default test cases on the website: test to see if the program runs as it should
   2. String names[10]= {“arnold”, “chad”, “chad”, “davidsmallberg”, “icarus”, “dwaynetherockjohnson”, “kyoto”, “kyoto”, “chad”, “chad”}
      1. separate(names, 7, “b”): test strange cases for separate
      2. separate(names, 4, “weirdalyankovich”): more strange cases
      3. separate(names, 9, “icarus”): see what happens when separator equals a value in the string
      4. separate(names, 10, “”): blank comparison value
      5. flip(names, 0): check flip function for 0 flip
      6. countRuns(names, 10): should return 7
      7. countRuns(names, 0): what happens for countRuns 0?
      8. rotateLeft(names, 10, 10): testing a “removal” of last position value
      9. rotateLeft(names, 0, 10): should return -1
      10. positionOfMax(names, 0): size of 0, should return -1
      11. lookup(names, 10, “chad”): should return smallest position where chad is found
      12. appendToAll(names, 10, “ “): what happens when we add spaces?
      13. appendToAll(names, 2, “\*&(\*^$#F”): adding junk
   3. string secondArray[10]= {“arnold”, “chad”, “chad”, “davidsmallberg”, “skywalker”}
      1. differ(names, 10, secondArray, 5): check if differ works for different values
      2. differ(names, 2, secondArray, 5): first array smaller than the second
      3. differ(names, 0, secondArray, 0): both arrays 0
      4. differ(names, 0, secondArray, 0): first array 0
      5. differ(names, 10, secondArray, 0): second array 0
   4. string subsequenceTester[10]= {“chad”, “chad”}
      1. subsequence(names, 10, subsequenceTester, 2): check function to see if it works
      2. subsequence(names, 10, subsequenceTester, 1): should return 4 instances of chad
   5. string lookUpStuff[10]= {“kyoto”, “davidsmallberg”}
      1. lookupAny(names, 10, lookUpStuff, 2): test function with new values
      2. lookupAny(names, 10, lookUpStuff, 1): look up a later value in the first array, should return a larger value