Jimmy Gieseke

CSC 201

Report

Lab 3 report

The objectives of this program were mostly working with strings and loops to look through multiple strings and find matches, as well as find common strings within two strings.

In my code I used a few variables DNA represented the first DNA string, and DNA2 represented the second when needed, as well as defining some functions to make the final code look a little neater. I also used A to hold a variable to follow through if a DNA string was Valid or Invalid (1 for valid and nothing for invalid) and also to hold something to be printed. For the references I used Ref1, Ref2, and Ref3 to hold the reference list.

In the final project (7) my code starts by defining the constants, the reference list, the DNA sequence input by the user, and the Valid DNA characters, and creates a new list for the DNA matches. The program then defines the bestmatch function witch will be used later. Then the code checks to see if the code is valid or invalid, if all indexes of DNA are True, we assign A the value 1, and if not we print the DNA is invalid and the code will have nothing to continue. Assuming the sequence was Valid we then move into the next if loop. When A is 1 we then run the function “bestmatch” defined earlier 3 times, with each Reference sequence. Each loop through we add the longest substring into the list we created at the very beginning. Then using this new list of 3 three strings, we can run through, and if the longest string was in index 0, it was a part of reference 1, if the longest string was part of index 1 the best match is reference 2, and if the longest string was part of index 2 of lsDNAmatch then the best match is reference 3.

Some issues I had with this program was figuring out how to find the longest common substring, counting matches was easy, but finding the longest set I had a lot of trouble with, and as I was about to submit the assignment, I realized that it would output an answer but that answer would not be the longest string, just the first string causing me to redo project 6 and 7. Another issue I had with the program was then trying to figure out how to run the program with three different DNA2s, luckily we learned about functions so I could figure out how to run it 3 times without the program being massive.

Some issue my code has is that if your DNA sequence has more than one match of the same amount of letters, the output will be the first reference, because say a list of (XXX, X, XXX) has the maximum of 3 in two situations, it will output the best solution as Ref1 since it comes first in the If loop, and I am unsure how to write it so it has multiple outputs of Ref1 and Ref3 as the best solutions.