Alex Chen, CSC-201

Lab03

Report03

The object of this program, part one, is to create codes to mirror DNA molecules, invert the molecules, validate them, combine the projects into one whole program and do all of the above. For part two, we had to match the given DNA sequences with the best outcome, finding the longest consecutive matching section between the sequences and then finally putting them all together and make it work as whole program.

Important variables are: DNA, a variable I used to store the input of a string to use and match the function of the code.

newDNA, a variable I used to store the input of the new string that the code comes out with.

valid\_DNA, a variable I used to store the letters of the nitrogenous bases, AGTC, and make sure the input of the user matches it. Otherwise the input by the user would be invalid.

DNA1, DNA2, are two variables I used to store the inputs of the user to use and match the function of the code.

dna\_Frag, a variable used to store the input of the string to use and match the function of the code.

gene1through4, variables storing the gene database for the code to be compared with.

Gene1-4match, storing the function, dna\_Match, to find and match between the user input and the gene database.

Gene1-4length, storing the function, dna\_Length, to find and match the longest consecutive sequence between the users input and the gene database.

Matchlist, is storing a list of the variables, gene1-4match, to then be used in a function to compare all the variables and finding the best matching DNA sequence.

Lengthlist, is storing a list of the variables, gene1-4length, to then be used in a function to compare all the variables and finding the best, longest matching DNA sequence.

Glist, a variable storing the function, compare, and then print out the results of the function.

Llist, a variable storing the function, compare, and then print out the results of the function.

How my programs operate: project 1 is to take the users input and then mirror each index of the string. Ex. A = T, T = A, C = G, G = C.

Project 2 takes the user input and then inverts the string. Ex. AGGT = TGGA

Project 3 takes the input and validates whether or not if it’s valid or invalid.

Project 4 is the combination between of projects 1, 2, and 3.

Project 5 is to take two user inputs and find which is the bigger string and smaller string. Then find the matching letters between the two strings.

Project 6 is to take the two inputs and find which is the bigger string and smaller string. Then find the longest consecutive matching sequence between the two.

Project 7 is the combination of 3, 5, and 6 with an additional part of comparing the user input against my created DNA database.