​In this project there were different tasks to calculate if the attribute contains valid bases,comparisions of the dna sequences, finding the pair of non-matching bases, reading data from the .dat files and printing total number of bases, finding the separators in the files and ploting the length of genes as a bar charts and making a scatter plot of the number of swap mutations per gene against gene length.

Task 1 :

In this task we have to check if seq contains valid bases. For this we have to write code in isdna method which returns whether seq contains valid bases and then after checking that returns True otherwise False.So we use a loop which checks the length of seq, if seq not in self bases then it passes False other wise the function will return passes. So for testing purpose if we uncomment the line myseq=dna.dnasequence(seq="U") in the project.py file and comment the myseq=dna.dnasequence(seq=mystr) in the same file then it will show our desired result. So if we do this it will return output “seq contains invalid bases. Exiting program.”

pseudo code :

if sequence contains valid bases

returns true

else

seq not in bases

then returns false

Task 2 :

In this task we have to make a method which will returns the complement of dna sequences. For this purpose I have made a method name complement which will return the complement of dna sequence and it will also return the comparison of two strings by using isequal method which is already defined. So it will compare the two strings name crick\_str and watson\_str, if they are complements then it will return True and if they are not complement it will return False.

Pesudo code:

Two strings crick\_str and watson\_str

comparasion between both

if both are complements

returns True

else

returns False

Task 3 :

In this task we have to create a method for finding the first pair of non-matching bases in 2 sequences. So I have created a method in dnasequence file named compare and in this method I am checking if the bases are of equal length.

If they are of equal length the method should return the index of mismatching bases. If they are not of equal length then the method is terminating the program and if method did not find any mismatch then it is returning -1. So for this task I have used if else statements and the output of first non matching bases in crick\_str and crick2\_str is 27.

Pesudo code:

if length of crick\_str is not equal to crick\_str2

then print “sequences are of different length “

and system exits

if both are of same length

prints “index at mismatch of bases”

if no mismatch found

then returns -1

Task 4 :

In this task we have to have to calculate the total number of bases in the genome\_01 file. So for this I have added a new function in the project.py file named def read(). This function reads the file of genome\_01 and then prints the length of total number of bases this file contains.

Pesudo code:

reading data from file genome\_01.dat

prints “total number of bases contains”

Task 5 :

In this task we have to first read the file genome\_01 and then find the separator “AAAAAAAAAATTTTTTTTTT”. After this it prints the length of first gene and plot the lengths as bar charts for the rest of genes. So for this I have made a method name gene() in the dnasequences.py file which first converts the string into tostring() and then find the separator in the genome\_ 01 file and then returns the length of first gene. After this it creates the list of dna sequences in the genome\_01 file and plot the length of the genes as a bar charts.

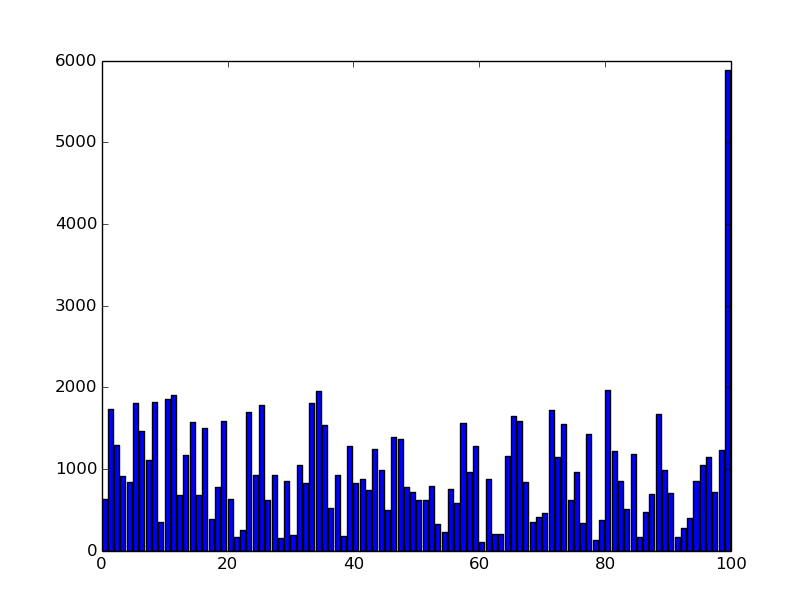
Pesudo code:

reading data from the file genome\_01.dat

finds separator “AAAAAAAAAATTTTTTTTTT”

then prints “length of first gene”

and plots

“the length of all gene as a bar charts”

Task 6 :

In this task we have to read two files name genome\_01 and genome\_02. After reading both files we have to count the number of non-equal bases in each gene. For this I have made a function name find\_nonmatch(). It will count all the non equal bases in each gene and also make a scatter plot of the number of swap mutations per gene against gene length.

Pseudo code:

reading data from the files genome\_01.dat and genome\_02.dat

then “counts the number of non-equal bases”

and plots

“number of swap mutations”

