1. Hours to complete the project:~ 15-20 hours 2. Short description of how you approached each problem, issues you encountered, and how you resolved those issues. Problem 1 (Markov Model Data Type):~ The first problem asked us to create a data type, representing the Markov model , from a text. Inorder to create a data type of Markov model we had to implement an API, that holds the constructer and other methods of the Markov Model data type. The first issue I encountered was to create a dictionary that holds the string of characters as its key and a dictionary as it values. Doing this was not an issue but generating the dictionary inside which holds the keys as letters and values as freqeuncy was a little tricky. After trying different ways in the interactive python, first I created the inside dictionary with the keys as the letters following the string, and then assigned each value with 0. Then going inside the second dictioanry I increased the value of the specific letter that follows the string.After that other methods were not that tricky. The rand and gen methods were a little confusing but after a little time on the interactive python I had no issues with those. I did not approach the replace_ unkown method here. After reading the question I felt like leaving it for the third problem since that is the client to test it. Problem 2 (Random Text Generator):~ The second problem asked us to write a client program that would read the text from the user and create a markov model data type that can generate the characters form the text. This was a simple problem as we only had to create a data type of markov model and print the data produced from the gen method. Problem 3 (Noisy Message Decoder):~ The third problem to we had to write the client program that would print out the most likely orginal string. The code in the client was not that bad because all we had to was take two command line arguments, create a markov model data type, and write the unknown letters. To write the unknown letters we had to call replace_unknown and here was the issue. It was very hard to understand how the replace unkown would work, but after talking to the professor, TAs and SI leader I figured out how to find, which character has the highest probablity of reapeating the text before it. We had to replace the value and check whether that value has the higher probality than the other or not. And using the argmax function we can find the value with the highes probality. 3. Did you receive any help from classmates, past CS110 students, TAs, or anyone else? Please list their names:~ Prof. Iyer, SI leader Michiele and TA Ryan 4. Other comments:~ Starting the other project right now

[illegible]