Jacob Reed

Lab #3

CS 408

**Three Phase Cipher**

**Purpose:** The purpose of this application is to apply the three phase cipher. This was accomplished by dividing the program’s methods up into three different groups for each phase of encryption and decryption. This would make it easier to process the information from each of the steps as it was handed off to the next phase.  
  
**main(String args[])**

This method is the main method of the program that is responsible for running each of the

three phases. Based on the flag provided (1, 2, 3) it would then execute the dictated function of encrypting, decrypting, or both operations.

**private static void codec(String arg)**

Allows for the argument to be passed in and based on the chosen operation it would then collect the text to be decoded or encoded, take in the appropriate values for the different portions of the key.

**ENCRYPTION**

**private static boolean checkNVals(int nValIn)**

This checks each of the values for the second part of the key making sure that they are within the correct range. If they are not then it will return false which will trigger a loop in a separate portion of the program to prompt the user again.

**private static void phaseOneEnc(String plainTextIn)**

This first phase will add the value of X to each of the letter’s ASCII values and store those values in an array to be used later.

**private static void phaseTwoEnc(int[] asciiWithX)**

This next phase will add the second portion of the key to the previous items. It will loop back around to the first part of the key when it reaches the end of the second part until each of the values from the previous phase have part two of the key added to them.

**private static void phaseThreeEnc(int[] phaseTwoResult)**

This method of the program will create a new bit string of the values from the results of the last phase. Once the bit string has been formed, the string will be divided again into the appropriate size that corresponds with the block size of **Y**. These values will also be stored in an array. The XOR operations will be next first with the Y bitstring with the first portion of the pre-CT bitstring and then the next block of the pre-CT bitstring with the previous result and so on till the end of the pre-CT bitstring. Lastly, additional zeros are added to the end to make sure that it can be evenly divided by eight.

**private static void bitStringToCT(String bitString)**

This portion of the code will take in the bitstring result from the third phase of the encryption and change it into the ASCII values for the cipher text. First, the bit string is divided into blocks of eight bits. Next, the bits will be converted into the appropriate char by parsing block into an int and get the ASCII character that corresponds to a character.

**DECRYPTION**

**private static void phaseOneDec(String cipherText)**

This will operate similar to the third phase of the encryption but will reverse the entire process and then outputting the decrypted bit string which would be the same as what was output at the end of phase two of the encryption.

**private static void phaseTwoDec(String bitString)**

This will also be a reverse process in relation to phase two and one. This will contain similar processes like the first two operations. The output from this portion of the decryption will have the result that is the same as the output from the first phase of the encryption where all of the appropriate key values and and X were added.

**private static void phaseThreeDec(int[] phaseTwoRes)**

This will take the result from the end of the phase two of the decryption and will remove the key values and the X from the blocks. This result will be output. Next, the blocks will be changed back to the ASCII values to get the plaintext back which will be the original message.