|  |  |
| --- | --- |
|  | TEST CASES |
| BELLASO CYPHER PASS |  |
| BELLASO CYPHER FAIL |  |
| CESAR CYPHER PASS |  |
| CESAR CYPHER FAIL |  |

|  |  |
| --- | --- |
| PSEUDO CODE | JAVA CODE |
| Method 1  The for loop loops throught all of the character in plain text to check if they are in valid bounds  Checks if plain text is in valid bounds  Returns false  Returns true | public static boolean stringInBounds (String plainText)  {    for (int i = 0; i < plainText.length(); i++)  {  if(plainText.charAt(i) < LOWER\_BOUND || plainText.charAt(i) > UPPER\_BOUND)  {  return false;  }  }    return true;    } |
| Initializes result  Loops through plain text  Initializes temp and sets it equal to the result of the encryption  Checks is temp is within the valid range  Sets result equal to result + temp  Returns result | public static String encryptCaesar(String plainText, int key) {    String result ="";    for (int i=0; i<plainText.length(); i++)  {  int temp = plainText.charAt(i) + key;    while(temp > UPPER\_BOUND)  {  temp -= RANGE;  }    result += (char)temp;    }  return result;  } |
| Initializes string encrypted string to nothing  Loops through plain text  Initializes temp and sets it to plainText plus bellasoStr  Validates if temp is within valid bounds  Adds temp to encrypted string  Returns encrypted string | public static String encryptBellaso(String plainText, String bellasoStr)  {  String encryptedString = "";    for (int i=0; i< plainText.length(); i++)  {  int temp = plainText.charAt(i) + bellasoStr.charAt(i % bellasoStr.length());    while (temp > UPPER\_BOUND)  {  temp-= RANGE;  }    encryptedString += (char) temp;    }  return encryptedString;    } |
| Initializes result  Loops through encrypted text  Initialized temp to encrypted text - key  Validates if the temp is within valid bounds  Adds temp to the result variable  Returns result | public static String decryptCaesar(String encryptedText, int key)  {  String result ="";    for (int i=0; i<encryptedText.length(); i++)  {  int temp = encryptedText.charAt(i) - key;    while(temp < LOWER\_BOUND)  {  temp += RANGE;  }    result += (char)temp;  }  return result;    } |
| Initializes the decryptedString  Loops through encrypted text    Initializes temp and sets it to plainText minus  bellasoStr  Checks if temp is within valid bounds  Adds temp to decryptedString  Returns decryptedString | public static String decryptBellaso(String encryptedText, String bellasoStr) {    String decryptedString ="";    for (int i=0; i<encryptedText.length(); i++)  {  int temp = encryptedText.charAt(i) - bellasoStr.charAt(i % bellasoStr.length());    while (temp <LOWER\_BOUND)  {  temp += RANGE;  }    decryptedString += (char) temp;    }  return decryptedString;  } |

Learning Experience

I learned how to use Junit tests, I used the Junit tests to test my program and to check if there are any errors within each specific method. I also furthered my knowledge in using if statements , while loops, for loops and using boolean variables. I also am more comfortable using and following methods and variables that get passed into the methods.