1. Create a method stringInBounds which takes a string plainText and returns a Boolean value
2. Initialize a Boolean ok to true
3. Start a for loop (int i=0; i<plainText.length(); i++)
4. if (plainText.charAt(i) < LOWER\_BOUND || plainText.charAt(i) > UPPER\_BOUND), return false
5. close for loop
6. return true
7. Create a method encryptCaesar which takes a string plainText and an int key and returns a string value
8. Initialize a string encryptedText to ""
9. if(stringInBounds(plainText)), Start a for loop for (int i=0; i<plainText.length(); i++)
10. initialize a char thisChar to plainText.charAt(i)
11. initialize an int encryptedCharInt to ((int)thisChar+key)
12. while (encryptedCharInt>UPPER\_BOUND), subtract encryptedCharInt by RANGE
13. increase encryptedText by (char)encryptedCharInt
14. close for loop
15. return encryptedText
16. Create a method encryptBellaso which takes two strings plainText and bellasoStr and returns a string value
17. Initialize a string encryptedText to ""
18. Initialize int bellasoStrLength to bellasoStr.length()
19. if(stringInBounds(plainText)), Start a for loop for (int i=0; i<plainText.length(); i++)
20. initialize a char thisChar to plainText.charAt(i)
21. initialize an int encryptedCharInt to ((int)thisChar+(int)bellasoStr.charAt(i %bellasoStrLength))
22. while (encryptedCharInt>UPPER\_BOUND), subtract encryptedCharInt by RANGE
23. increase encryptedText by (char)encryptedCharInt
24. close for loop
25. return encryptedText
26. Create a method decryptCaesar which takes a string encryptedText and an int key and returns a string value
27. Initialize a string decryptedText to ""
28. Start a for loop for (int i=0; i< encryptedText.length(); i++)
29. initialize a char thisChar to encryptedText.charAt(i)
30. initialize an int decryptedCharInt to ((int)thisChar-key)
31. while (decryptedCharInt<LOWER\_BOUND), increase decryptedCharInt by RANGE
32. increase decryptedText by (char) decryptedCharInt
33. close for loop
34. return dencryptedText
35. Create a method dencryptBellaso which takes two strings encryptedText and bellasoStr and returns a string value
36. Initialize a string dencryptedText to ""
37. Initialize int bellasoStrLength to bellasoStr.length()
38. Start a for loop for (int i=0; i<encryptedText.length(); i++)
39. initialize a char thisChar to encryptedText.charAt(i)
40. initialize an int dencryptedCharInt to ((int)thisChar-(int)bellasoStr.charAt(i %bellasoStrLength))
41. while (dencryptedCharInt<LOWER\_BOUND), increase dencryptedCharInt by RANGE
42. increase dencryptedText by (char)dencryptedCharInt
43. close for loop
44. return dencryptedText

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input text | Input Key | Encrypted (method1) | Encrypted (method2) | Decrypt (method1) | Decrypt (method2) |
| HELLO | 5 | MJQQT |  | HELLO |  |
| NAMASTE | 8 | VIUI[\M |  | NAMASTE |  |
| KABINDRA | POTATO |  | [PVJ"S"P |  | KABINDRA |
| COMPUTER | SCIENCE |  | VRVU#WJ% |  | COMPUTER |