Caesar Cipher in Cryptography

The Caesar Cipher technique is one of the earliest and simplest method of encryption technique. It’s simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter some fixed number of positions down the alphabet.

For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.  
Thus to cipher a given text we need an integer value, known as shift which indicates the number of position each letter of the text has been moved down.  
The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically as.





**Examples :**

**Text** : ABCDEFGHIJKLMNOPQRSTUVWXYZ

**Shift**: 23

**Cipher**: XYZABCDEFGHIJKLMNOPQRSTUVW

**Text** : ATTACKATONCE

**Shift**: 4

**Cipher**: EXXEGOEXSRGI

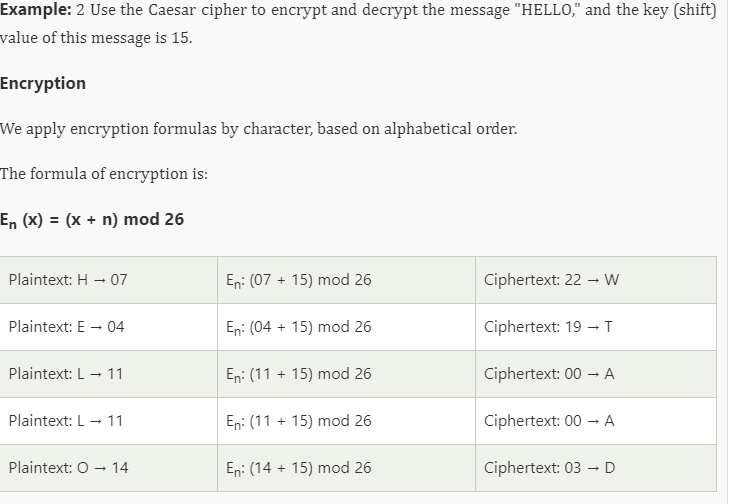
**Algorithm for Caesar Cipher:**  
**Input:**

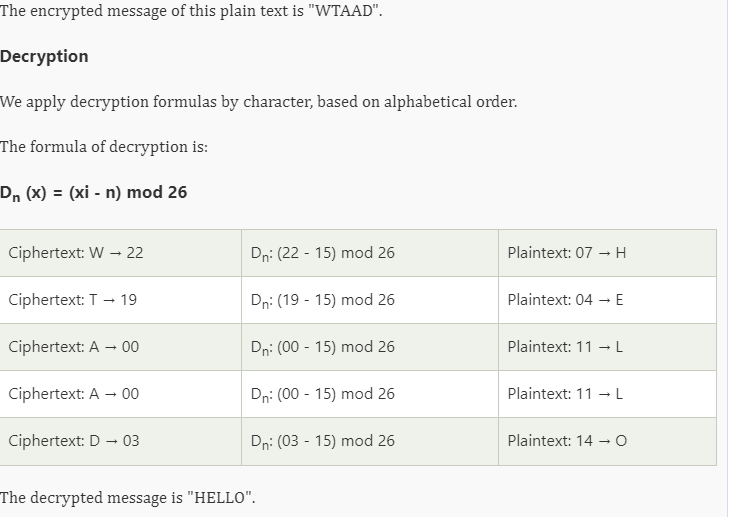
1. A String of lower case letters, called Text.
2. An Integer between 0-25 denoting the required shift.

**Procedure:**

* Traverse the given text one character at a time .
* For each character, transform the given character as per the rule, depending on whether we’re encrypting or decrypting the text.
* Return the new string generated.

Program that receives a Text (string) and Shift value( integer) and returns the encrypted text.





|  |
| --- |
| //A Java Program to illustrate Caesar Cipher Technique  class CaesarCipher  {      // Encrypts text using a shift od s      public static StringBuffer encrypt(String text, int s)      {          StringBuffer result= new StringBuffer();            for (int i=0; i<text.length(); i++)          {              if (Character.isUpperCase(text.charAt(i)))              {                  char ch = (char)(((int)text.charAt(i) +                                    s - 65) % 26 + 65);                  result.append(ch);              }              else              {                  char ch = (char)(((int)text.charAt(i) +                                    s - 97) % 26 + 97); 0                  result.append(ch);              }          }          return result;      }        // Driver code      public static void main(String[] args)      {          String text = "ATTACKATONCE";          int s = 4;          System.out.println("Text  : " + text);          System.out.println("Shift : " + s);          System.out.println("Cipher: " + encrypt(text, s));      }  } |

**o/p**

Text : ATTACKATONCE

Shift : 4

Cipher: EXXEGOEXSRGI

# \

# 

# 