SurajSG23 Create Affine Cipher 9f9befe · now

88 lines (74 loc) · 2.71 KB

Code Blame Raw

```java
import java.util.Scanner;

public class AffineCipher {

    // Function to perform modular inverse
    public static int modInverse(int a, int m) {
        a = a % m;
        for (int x = 1; x < m; x++) {
            if ((a * x) % m == 1) {
                return x;
            }
        }
        return -1;
    }

    // Encrypt the plaintext using Affine Cipher
    public static String encrypt(String text, int a, int b) {
        StringBuilder cipherText = new StringBuilder();
        text = text.toLowerCase();
        for (int i = 0; i < text.length(); i++) {
            char c = text.charAt(i);
            if (c >= 'a' && c <= 'z') {
                int x = c - 'a';
                char encryptedChar = (char) (((a * x + b) % 26) + 'a');
                cipherText.append(encryptedChar);
            } else {
                cipherText.append(c); // Keep non-alphabetic characters unchange
            }
        }
        return cipherText.toString();
    }

    // Decrypt the ciphertext using Affine Cipher
    public static String decrypt(String text, int a, int b) {
        StringBuilder plainText = new StringBuilder();
        int a_inv = modInverse(a, 26); // Find modular inverse of a
```

```
37
38              if (a_inv == -1) {
39                  return "Inverse of 'a' does not exist.";
40              }
41
42              text = text.toLowerCase();
43              for (int i = 0; i < text.length(); i++) {
44                  char c = text.charAt(i);
45                  if (c >= 'a' && c <= 'z') {
46                      int y = c - 'a';
47                      char decryptedChar = (char) (((a_inv * (y - b + 26)) % 26) + 'a'
48                      plainText.append(decryptedChar);
49                  } else {
50                      plainText.append(c); // Keep non-alphabetic characters unchanged
51                  }
52              }
53              return plainText.toString();
54          }
55
56          public static void main(String[] args) {
57              Scanner scanner = new Scanner(System.in);
58
59              System.out.println("Affine Cipher");
60
61              // Get inputs for encryption
62              System.out.print("Enter the plaintext: ");
63              String plaintext = scanner.nextLine();
64              System.out.print("Enter the multiplier (a): ");
65              int a = scanner.nextInt();
66              System.out.print("Enter the shift (b): ");
67              int b = scanner.nextInt();
68
69              // Encrypt the plaintext
70              String encryptedText = encrypt(plaintext, a, b);
71              System.out.println("Encrypted Text: " + encryptedText);
72
73              // Decrypt the ciphertext
74              String decryptedText = decrypt(encryptedText, a, b);
75              System.out.println("Decrypted Text: " + decryptedText);
76
77              scanner.close();
78          }
79      }
80
81
82      //Output
83      Affine Cipher
84      Enter the plaintext: hello i am Suraj
85      Enter the multiplier (a): 1
86      Enter the shift (b): 4
87      Encrypted Text: lipps m eq wyven
88      Decrypted Text: hello i am suraj
```