

Problem Statement: "To design a Affine cipher"

Code :

```
/*Name: ANISH MORE
Registration number: 2241013365
Section: 2241037
Branch: CSE
Name of the program : To design a Affine cipher.
*/
import java.util.Scanner;
public class Affinecipher
{
    static int k1;
    static int k2;
    public static void getKeyValue()
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the value of first key(k1) : ");
        k1=sc.nextInt();
        System.out.print("Enter the value of second key(k2) : ");
        k2=sc.nextInt();
    }
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the message : ");
        String plaintext=sc.nextLine();
        getKeyValue();
        String ciphertext=encrypt(plaintext);
        System.out.println("After encryption, the ciphertext is : "+ciphertext);
        String plaintext_after_decryption=decrypt(ciphertext);
        System.out.println("After decryption, the plaintext is : "+plaintext_after_decryption);
    }
    public static String encrypt(String plaintext)
    {
        String ciphertext="";
        for(int i=0;i<plaintext.length();i++)
        {
            char character=plaintext.charAt(i);
            if(character!=' ')
            {
                int value=0;
```

```

        for(int j=0;j<26;j++)
        {
            if(((char)(97+j))==character)
            {
                value=j;
                break;
            }
        }

        int C=(value*k1+k2)%26;
        ciphertext=ciphertext+((char)(65+C));
    }

    else
    {
        ciphertext=ciphertext+character;
    }
}
return ciphertext;
}

public static String decrypt(String ciphertext)
{
    String plaintext_After_Decryption="";
    for(int i=0;i<ciphertext.length();i++)
    {
        char character=ciphertext.charAt(i);
        int value=-1;
        if(character!=' ')
        {
            for(int j=0;j<26;j++)
            {
                if((char)(65+j)==character)
                {
                    value=j;
                    break;
                }
            }

            int additive_inverse=Math.abs(26-k2)%26;
            int multiplicative_Inverse=multiplicative_inverse(k1);
            if (multiplicative_Inverse== -1)
            {

```

```

        System.out.println("No multiplicative inverse found for k1 as k1 is even number
so decryption cannot be done and it will return null. " );
        return null;
    }
    int P=((((value+additive_inverse)%26)*multiplicative_Inverse)%26)%26;
    plaintext_After_Decryption=plaintext_After_Decryption+(char)(97+P);
}
else
{
    plaintext_After_Decryption=plaintext_After_Decryption+character;
}
}
return plaintext_After_Decryption;
}
public static int multiplicative_inverse(int k1)
{
    k1= k1%26;
    for (int i = 1; i < 26;i++)
        if ((k1 * i) % 26 == 1)
            return i;
    return -1;
}
}

```

Description :

Here 5 methods have been used -

- 1) public static void main(String[] args)
- 2) public static String encrypt(String plaintext)
- 3) public static String decrypt(String ciphertext)
- 4) public static int multiplicative_inverse(int k1)
- 5) public static void getKeyValue()

- **public static void main(String[] args)**

- **public static void getKeyValue()**

In main method, user is asked to enter the message which is to be encrypted and decrypted. The entered message is stored in variable 'plaintext'. Then it calls getKeyValue() to get the value of key from user. In getKeyValue method, the user enters the value of first key (k1) and second key(k2). The main method calls encrypt method(encrypt(String plaintext))to encrypt the message and display it on the console. The main method calls decrypt method decrypt(String ciphertext) to decrypt the message and display it on the console.

- **public static String encrypt(String plaintext)**

In encrypt method , a string named "ciphertext" is made to store the encrypted message. A for loop is used to traverse each alphabet present in the message(plain text) and inside the

loop it will check if the alphabet is equal to space or not. If it is not a space then inside this another for loop is used to get the value or index of the alphabet. After getting the value that alphabet is encrypted. Like this it continues till all the alphabet is encrypted. Then this encrypt method returns the value of the ciphertext to main method to print it in console.

- **public static String decrypt(String ciphertext)**

In decrypt method, a string named "plaintext_After_Decryption" is made to store the plaintext after decrypting the encrypted message(cipher text). A for loop is used to traverse each alphabet of ciphertext and inside the loop it check if the alphabet is equal to space or not. If it is not a space then inside this another for loop is used to get the value/ index of that alphabet. After getting the value the alphabet is decrypted and stored into plaintext_After_Decryption variable. To decrypt, it uses a method called multiplicative_inverse to get the inverse of key k1. It will check if multiplicative inverse value is -1 or not. -1 is used because the user can enter a value of k1 in even number whose multiplicative inverse cannot be found. If it multiplicative inverse is -1 it will return a error statement and null. The decryption method returns the value of plaintext_After_Decryption to the main method to print it in console.

- **public static int multiplicative_inverse(int k1)**

This method finds the multiplicative inverse of first key k1. Then check if $(k1 * i) \% 26 == 1$, then i will be the multiplicative inverse of k1.

Output:

Test Case 1:

Enter the message : hello

Enter the value of first key(k1) : 7

Enter the value of second key(k2) : 2

After encryption, the ciphertext is : ZEBBW

After decryption, the plaintext is : hello

Test Case 2:

Enter the message : anish

Enter the value of first key(k1) : 9

Enter the value of second key(k2) : 3

After encryption, the ciphertext is : DQXJO

After decryption, the plaintext is : anish

Test Case 3:

Enter the message : hello everyone

Enter the value of first key(k_1) : 7

Enter the value of second key(k_2) : 2

After encryption, the ciphertext is : ZEBBW ETEROWPE

After decryption, the plaintext is : hello everyone