```
import java.util.Scanner;

public class MultiplicativeCipher {


    public static String clean(String text) {
        return text.toLowerCase().replaceAll("[^a-z]", "");
    }

     modulo 26
    public static int multiplicativeInverse(int key) {
        key = key % 26;
        for (int i = 1; i < 26; i++) {
            if ((key * i) % 26 == 1)
                return i;
        }
        return -1; // (مفتاح غير صالح) لا يوجد معكوس
    }

    // التشفير: C = (P * K) mod 26
    public static String encrypt(String text, int key) {
        text = clean(text);
        StringBuilder cipher = new StringBuilder();

        for (char c : text.toCharArray()) {
            int p = c - 'a';
            int ciph = (p * key) % 26;
            cipher.append((char) (ciph + 'a'));
        }

        return cipher.toString();
    }

    // فك التشفير: P = (C * K_inv) mod 26
    public static String decrypt(String cipherText, int key) {
        cipherText = clean(cipherText);
        int kInv = multiplicativeInverse(key);

        if (kInv == -1)
            return "خطأ: المفتاح ليس له معكوس ولا يمكن فك التشفير!";

        StringBuilder plain = new StringBuilder();

        for (char c : cipherText.toCharArray()) {
            int ci = c - 'a';
            int p = (ci * kInv) % 26;
```

```
        plain.append((char) (p + 'a'));
    }

    return plain.toString();
}

// الهجوم
public static void bruteForce(String cipherText) {
    System.out.println("\n==== الهجوم الأعمى (Brute Force) ====\n");

    int[] validKeys = {1,3,5,7,9,11,15,17,19,21,23,25};

    for (int k : validKeys) {
        String attempt = decrypt(cipherText, k);
        System.out.println("المفتاح (" + k + ") → " + attempt);
    }
}


public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    System.out.println("=== المشفّر الجدائي Multiplicative Cipher ===");
    System.out.println("1) تشفير");
    System.out.println("2) فك تشفير");
    System.out.println("3) هجوم أعمى");
    System.out.print("اختر العملية: ");

    int choice = sc.nextInt();
    sc.nextLine();

    switch (choice) {
        case 1:
            System.out.print("أدخل النص الأصلي: ");
            String plain = sc.nextLine();
            System.out.print("أدخل المفتاح (من المفاتيح المسموحة فقط): ");
            int keyE = sc.nextInt();
            System.out.println("النص المشفّر: " + encrypt(plain, keyE));
            break;

        case 2:
            System.out.print("أدخل النص المشفّر: ");
            String cipher = sc.nextLine();
            System.out.print("أدخل المفتاح: ");
            int keyD = sc.nextInt();
```

```
            System.out.println("النص المفكوك: " + decrypt(cipher, keyD));
            break;

        case 3:
            System.out.print("أدخل النص المشفّر للهجوم عليه: ");
            String brute = sc.nextLine();
            bruteForce(brute);
            break;

        default:
            System.out.println("اختيار غير صالح!");
      }

      sc.close();
   }
}
```