**CNS Exp 4**

**Shashwat Tripathi  
D15A 64**

**Batch C**

**Aim:** Write a program in Java or Python to perform Cryptanalysis or decoding of Vignere Cipher

**Theory:**

Key Concepts:

* Polyalphabetic Substitution Cipher: Unlike monoalphabetic substitution ciphers (like Caesar cipher) that use a fixed substitution pattern for each letter, Vigenère cipher uses multiple substitution patterns based on a keyword. This makes it more secure compared to simple substitution ciphers.
* Keyword: The keyword is a secret sequence of characters (usually letters) that determines the shift value for each letter in the plaintext. It's the core of the Vigenère cipher and dictates the encryption and decryption process.
* Encryption: To encrypt a message using the Vigenère cipher, each letter of the plaintext is shifted based on the corresponding letter in the keyword. The shift value for a letter is determined by its position in the alphabet (A=0, B=1, ..., Z=25).
* Decryption: To decrypt an encrypted message, the reverse process is applied. Each letter of the ciphertext is shifted back based on the corresponding letter in the keyword.

Advantages: Vigenère cipher is stronger than simple Caesar cipher due to its use of multiple substitution patterns. It's relatively easy to understand and implement.

Disadvantages: The security of Vigenère cipher depends on the length and secrecy of the keyword. If the keyword is short or not truly random, it's susceptible to attacks such as frequency analysis. It's also vulnerable to Kasiski examination and Friedman test if the keyword length is too short.

**Code:**

public class VigenereCipher {

public static void main(String[] args) {

Vigenere cipher = new Vigenere("JAVA");

String decrypted = cipher.decrypt("BHVSQWVT");

System.out.println("Decrypted: " + decrypted);

}

static class Vigenere {

private String key;

public Vigenere(String key) {

this.key = "";

for (int i = 0; i < key.length(); ++i) {

char c = key.charAt(i);

if (c >= 'A' && c <= 'Z')

this.key += c;

else if (c >= 'a' && c <= 'z')

this.key += (char) (c + 'A' - 'a');

}

}

public String decrypt(String text) {

StringBuilder out = new StringBuilder();

for (int i = 0, j = 0; i < text.length(); ++i) {

char c = text.charAt(i);

if (c >= 'a' && c <= 'z')

c += 'A' - 'a';

else if (c < 'A' || c > 'Z')

continue;

out.append((char) ((c - key.charAt(j) + 26) % 26 + 'A'));

j = (j + 1) % key.length();

}

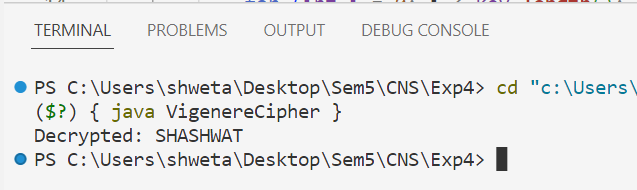
return out.toString();

}

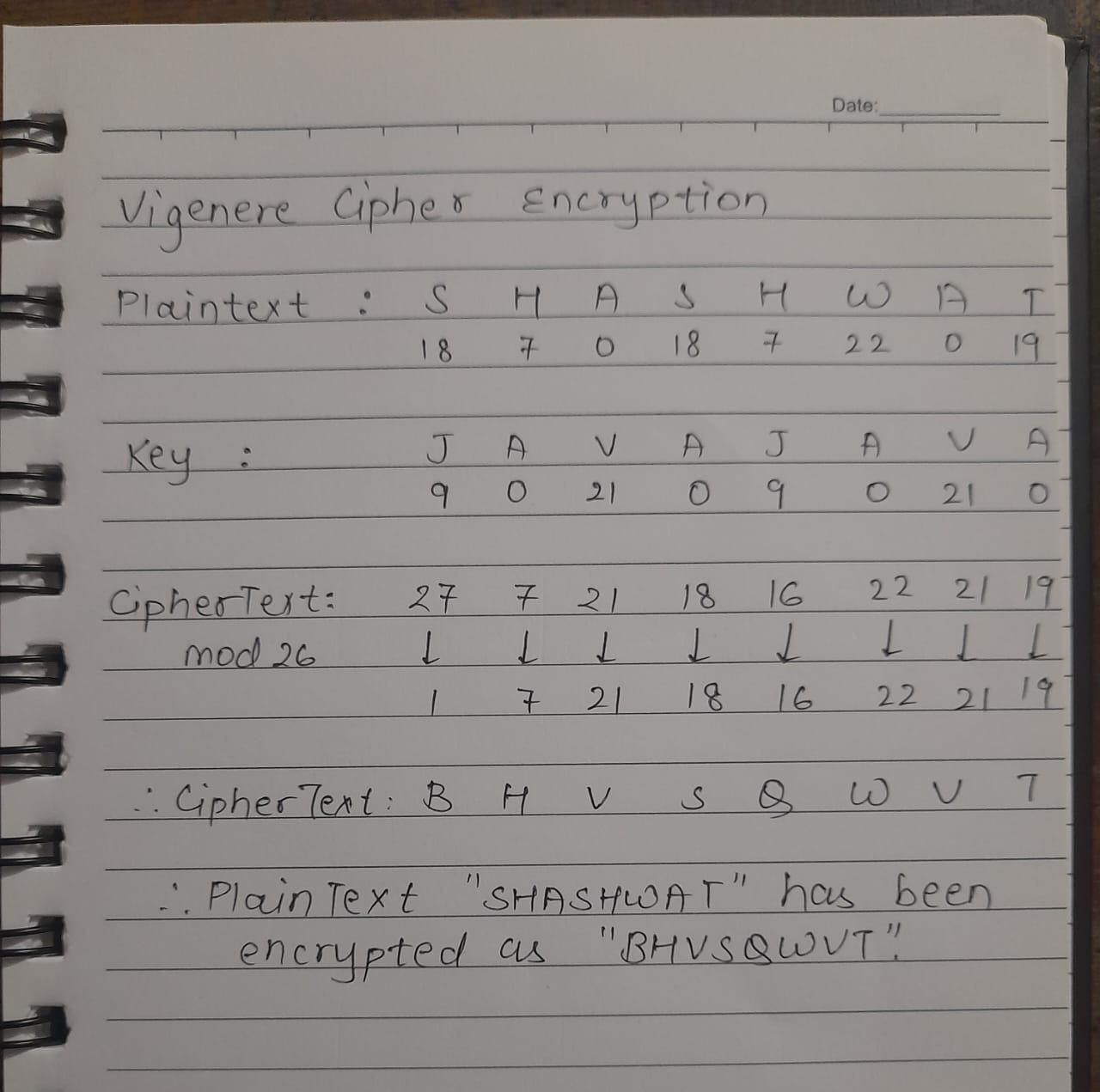
}

}

**Output:**

****

**Vigenere Cipher Encryption:**

****

**Conclusion:**

Thus, we have written and implemented a Java code to perform the decoding of Vignere Cipher.