**Experiment No 7 Implement and write advantages of Poly-alphabetic Cipher.** #include <iostream>

#include <string> using namespace std;

string encryptVigenere(string plainText, string key) { string encryptedText = "";

int keyLength = key.length();

for (int i = 0; i < plainText.length(); i++) { char plainChar = plainText[i];

char keyChar = key[i % keyLength]; // Repeating the key as necessary

if (isalpha(plainChar)) {

char base = isupper(plainChar) ? 'A' : 'a';

char encryptedChar = ((plainChar - base + keyChar - 'A') % 26) + base; encryptedText += encryptedChar;

} else {

encryptedText += plainChar; // Non-alphabet characters remain unchanged

}

}

return encryptedText;

}

string decryptVigenere(string encryptedText, string key) { string decryptedText = "";

int keyLength = key.length();

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

char encryptedChar = encryptedText[i]; char keyChar = key[i % keyLength];

if (isalpha(encryptedChar)) {

char base = isupper(encryptedChar) ? 'A' : 'a';

char decryptedChar = ((encryptedChar - base - keyChar + 26) % 26) + base; decryptedText += decryptedChar;

} else {

decryptedText += encryptedChar;

}

}

return decryptedText;

}

int main() {

string plaintext, key;

cout << "Enter the plaintext: "; getline(cin, plaintext);

cout << "Enter the key: "; cin >> key;

string encryptedText = encryptVigenere(plaintext, key); string decryptedText = decryptVigenere(encryptedText, key);

cout << "Encrypted Text: " << encryptedText << endl; cout << "Decrypted Text: " << decryptedText << endl;

return 0;

}

