Assignment 3

Personalized encryption/decryption algorithm

Code:-

```
#include <iostream>
#include <cctype>
#include <string>
using namespace std;
// Function to encrypt a message using the Polybius cipher
std::string polybiusEncrypt(const std::string &message)
{
    string encryptedMessage = "";
    string polybiusTable[5] = {
        "ABCDE",
        "FGHIJ",
        "KLMNO",
        "PQRST",
        "UVWXY"};
    for (char c : message)
    {
        if (isalpha(c))
        {
            c = toupper(c); // Convert to uppercase
            if (c == 'Z')
                c = 'Y'; // Handle 'Z' as 'Y'
            for (int row = 0; row < 5; ++row)
            {
                size_t col = polybiusTable[row].find(c);
                if (col != string::npos)
                {
                    encryptedMessage += std::to_string(row + 1) + std::to_string(col + 1);
                    break;
                }
            }
        }
        else
        {
            // Non-alphabetic characters are not encrypted
            encryptedMessage += c;
        }
    }
    return encryptedMessage;
}
// Function to decrypt a message using a reverse substitution cipher (Caesar cipher)
string decryptMessage(const std::string &encryptedMessage)
{
```

```
    std::string decryptedMessage = "";
    int shift = 1; // Caesar cipher shift value
    for (size_t i = 0; i < encryptedMessage.length(); ++i)
    {
        if (isdigit(encryptedMessage[i]))
        {
            int row = encryptedMessage[i] - '0';
            int col = encryptedMessage[++i] - '0' - 1;
            decryptedMessage += 'A' + (row - 1) * 5 + col;
        }
        else
        {
            // Non-digit characters are not decrypted
            decryptedMessage += encryptedMessage[i];
        }
    }
    return decryptedMessage;
}
int main()
{
    string message;
    cout << "Enter a message to encrypt: ";
    getline(cin, message);
    // Encrypt the message using the Polybius cipher
    string encryptedMessage = polybiusEncrypt(message);
    cout << "Encrypted message: " << encryptedMessage << std::endl;
    // Decrypt the message using a reverse substitution cipher (Caesar cipher)
    string decryptedMessage = decryptMessage(encryptedMessage);
    cout << "Decrypted message: " << decryptedMessage << std::endl;
    return 0;
}
```

Output:-

```
Enter a message to encrypt: Jayant Kumar
Encrypted message: 251155113445 3151331143
Decrypted message: JAYANT KUMAR
```

Conclusion:-

- The Polybius cipher is a straightforward substitution cipher that converts letters in the plaintext to coordinate pairs in a grid, producing ciphertext.
- While it is easy to implement and understand, it is not considered secure for modern cryptographic purposes due to its vulnerability to frequency analysis and the limited key space.
- In practice, more secure encryption methods like modern symmetric-key ciphers (e.g., AES) and asymmetric-key ciphers (e.g., RSA) should be used for protecting sensitive information.
- The Polybius cipher is suitable only for educational purposes or very basic encoding tasks.