Question # 1

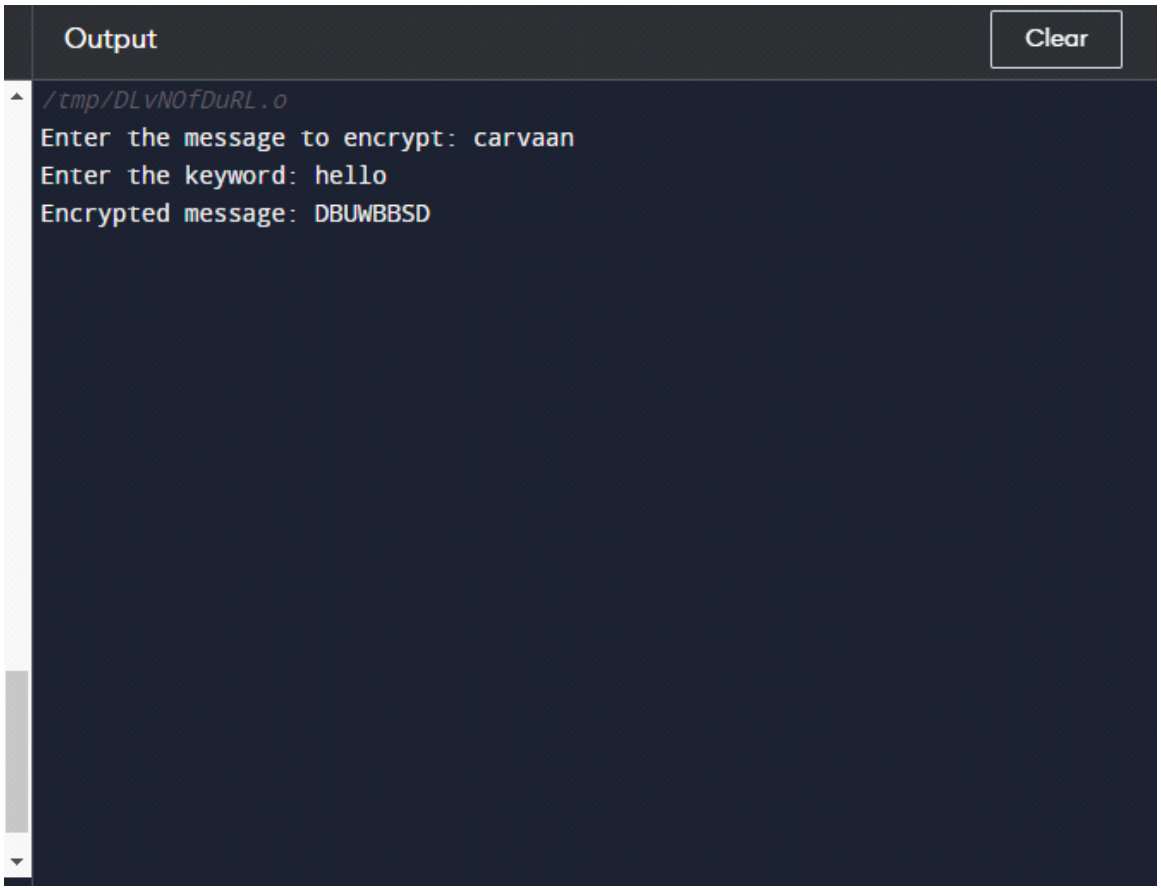
Write encryption code for playfair cipher in C++ .

```
#include <iostream>

#include <string>

#include <cctype>

using namespace std;


string prepareMessage(const string& message) {

	string preparedMessage;

	for (char ch : message) {

		if (isalpha(ch))

			preparedMessage += toupper(ch);

	}

	return preparedMessage;

}


void generateKeySquare(const string& keyword, char keySquare[5][5]) {

	string key = prepareMessage(keyword + "ABCDEFGHIJKLMNOPQRSTUVWXYZ");

	bool used[26] = { false };

	int row, col;


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

		char ch = key[i];

		if (!used[ch - 'A']) {

			used[ch - 'A'] = true;
```

```
                if (ch == 'J')
                        used['I' - 'A'] = true;

                row = i / 5;
                col = i % 5;
                keySquare[row][col] = ch;
            }
        }
}

void findPosition(const char keySquare[5][5], char ch, int& row, int& col) {
        if (ch == 'J')
            ch = 'I';

        for (row = 0; row < 5; ++row) {
            for (col = 0; col < 5; ++col) {
                if (keySquare[row][col] == ch)
                    return;
            }
        }
}

string encryptPlayfair(const string& message, const string& keyword) {
        char keySquare[5][5];
        generateKeySquare(keyword, keySquare);
```

```
string preparedMessage = prepareMessage(message);
string encryptedMessage;

for (int i = 0; i < preparedMessage.length(); i += 2) {
        char ch1 = preparedMessage[i];
        char ch2 = preparedMessage[i + 1];

        int row1, col1, row2, col2;
        findPosition(keySquare, ch1, row1, col1);
        findPosition(keySquare, ch2, row2, col2);

        if (row1 == row2) {
                encryptedMessage += keySquare[row1][(col1 + 1) % 5];
                encryptedMessage += keySquare[row2][(col2 + 1) % 5];
        }
        else if (col1 == col2) {
                encryptedMessage += keySquare[(row1 + 1) % 5][col1];
                encryptedMessage += keySquare[(row2 + 1) % 5][col2];
        }
        else {
                encryptedMessage += keySquare[row1][col2];
                encryptedMessage += keySquare[row2][col1];
        }
}

return encryptedMessage;
```

```
}

int main() {
    string message, keyword;

    cout << "Enter the message to encrypt: ";
    getline(cin, message);

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

    string encryptedMessage = encryptPlayfair(message, keyword);
    cout << "Encrypted message: " << encryptedMessage << endl;

    return 0;
}
```


Output
Clear
/tmp/DLvNOfDuRL.o
Enter the message to encrypt: carvaan
Enter the keyword: hello
Encrypted message: DBUWBBSD

Question # 2

Write decryption code for playfair cipher in C++ .

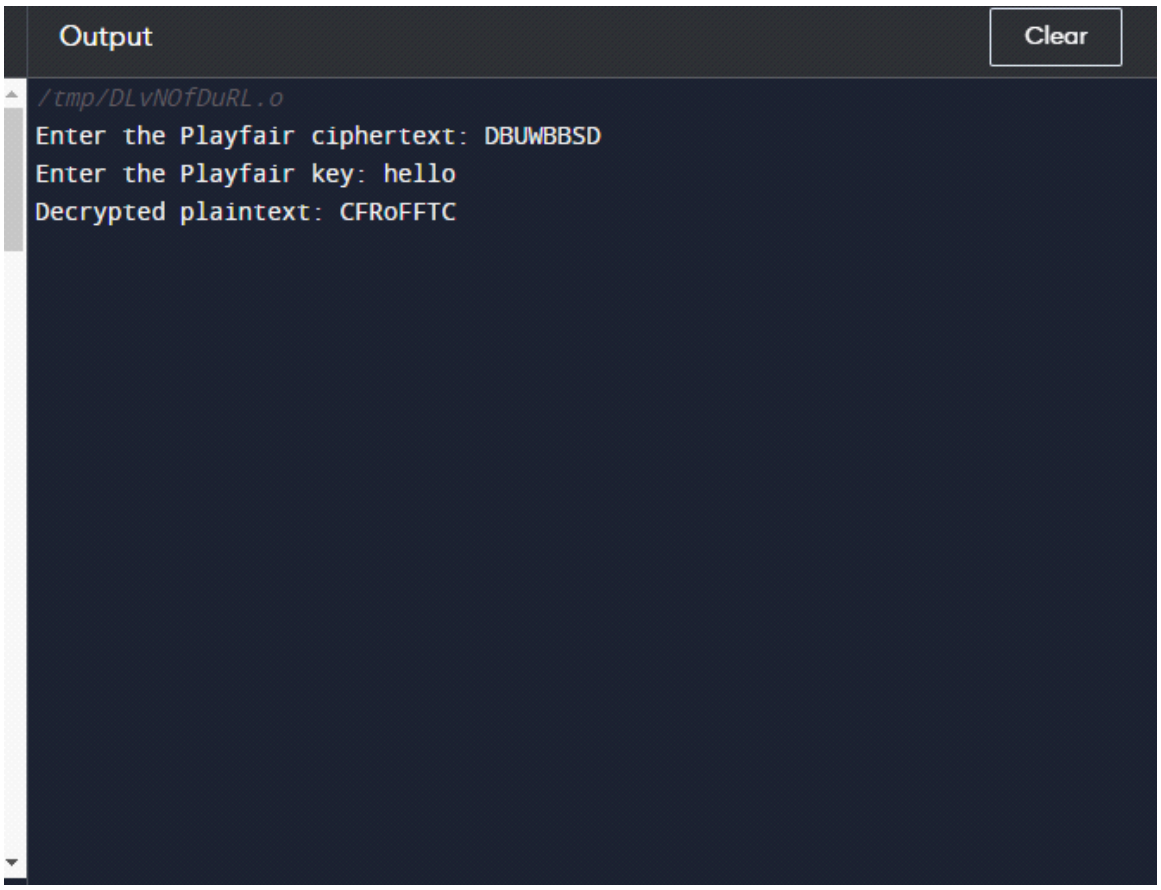
```
#include <iostream>

#include <string>

#include <vector>


using namespace std;


const string alphabet = "ABCDEFGHIKLMNOPQRSTUVWXYZ";


// Function to generate the Playfair cipher key

string generateKey(const string& key) {

	string playfairKey;

	vector<bool> visited(26, false);


	// Append the key to the playfairKey string

	for (char c : key) {

		if (c == 'J')

			continue; // 'J' and 'I' are treated as the same letter in Playfair cipher


		if (!visited[c - 'A']) {

			playfairKey += c;

			visited[c - 'A'] = true;

		}

	}
```

```
    // Append the remaining alphabet characters to the playfairKey string
    for (char c : alphabet) {
        if (!visited[c - 'A']) {
            playfairKey += c;
            visited[c - 'A'] = true;
        }
    }

    return playfairKey;
}

// Function to find the row and column of a character in the Playfair cipher grid
void findPosition(const string& playfairKey, char ch, int& row, int& col) {
    for (int i = 0; i < 5; i++) {
        for (int j = 0; j < 5; j++) {
            if (playfairKey[i * 5 + j] == ch) {
                row = i;
                col = j;
                return;
            }
        }
    }
}

// Function to decrypt a pair of characters using the Playfair cipher rules
```

```
string decryptPair(const string& playfairKey, char ch1, char ch2) {
    int row1, col1, row2, col2;
    findPosition(playfairKey, ch1, row1, col1);
    findPosition(playfairKey, ch2, row2, col2);

    string decryptedPair;

    if (row1 == row2) {
        col1 = (col1 - 1 + 5) % 5;
        col2 = (col2 - 1 + 5) % 5;
    } else if (col1 == col2) {
        row1 = (row1 - 1 + 5) % 5;
        row2 = (row2 - 1 + 5) % 5;
    } else {
        swap(col1, col2);
    }

    decryptedPair += playfairKey[row1 * 5 + col1];
    decryptedPair += playfairKey[row2 * 5 + col2];

    return decryptedPair;
}

// Function to decrypt the Playfair cipher text
string decryptPlayfair(const string& ciphertext, const string& key) {
    string playfairKey = generateKey(key);
```

```
    string decryptedText;

    for (int i = 0; i < ciphertext.length(); i += 2) {
        char ch1 = ciphertext[i];
        char ch2 = ciphertext[i + 1];
        decryptedText += decryptPair(playfairKey, ch1, ch2);
    }

    return decryptedText;
}

int main() {
    string ciphertext;
    string key;

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

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

    string plaintext = decryptPlayfair(ciphertext, key);

    cout << "Decrypted plaintext: " << plaintext << endl;

    return 0;
```

}


Output
Clear
/tmp/DLvNOfDuRL.o
Enter the Playfair ciphertext: DBUWBBSD
Enter the Playfair key: hello
Decrypted plaintext: CFRoFFTC