EX NO:02	Write a program for demonstration of encrypting and decrypting the messages by Playfair Substitution technique.
DATE:21.02.24	

Aim: To write and execute a program for demonstration of encrypting and decrypting the messages by Playfair Substitution technique.

Algorithm:

1. Define a constant SIZE to represent the size of the key table (5x5 matrix).
2. Create a function prepareKeyTable that takes the user-provided key and fills a 5x5 key table with the characters of the key and the remaining alphabet (excluding 'J').
3. Use nested loops to iterate over the rows and columns of the key table until the character is found.
4. Create a function playfair that takes the key table, the message, and an empty result array.
5. Declare variables for the key, message, result, and key table.
6. Call the prepareKeyTable function to generate the key table.
7. Call the playfair function to perform encryption/decryption and store the result.
8. Print the encrypted/decrypted message,
9. Prompt the user to enter the key and the message.
10. Print the encrypted/decrypted message using printf.
11. Compile the C program using a C compiler (e.g., gcc).
12. Run the compiled program.

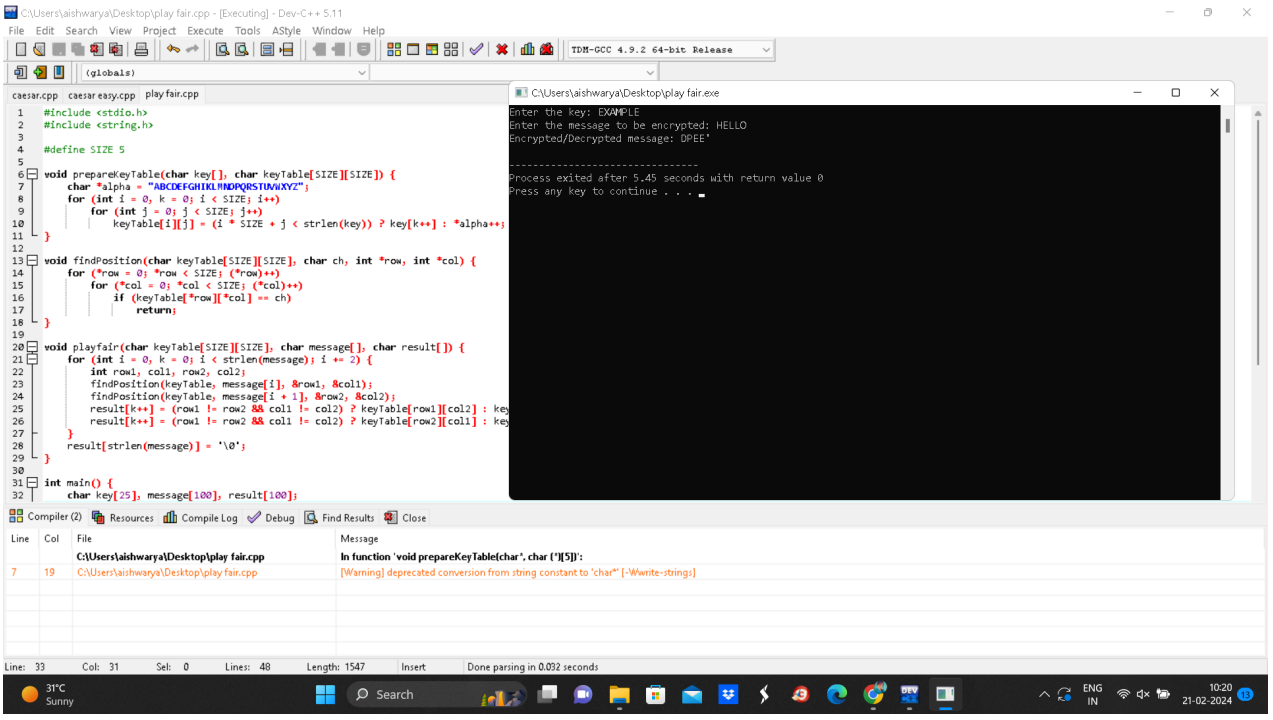
Program:

```
#include <stdio.h>

#include <string.h>

#define SIZE 5



// Function to prepare the Playfair Square key table

void prepareKeyTable(char key[], char keyTable[SIZE][SIZE]) {

   char *alpha = "ABCDEFGHIKLMNOPQRSTUVWXYZ";

   for (int i = 0, k = 0; i < SIZE; i++)

      for (int j = 0; j < SIZE; j++)

         keyTable[i][j] = (i * SIZE + j < strlen(key)) ? key[k++] : *alpha++;
```

```
}

// Function to find the position of a character in the Playfair Square
void findPosition(char keyTable[SIZE][SIZE], char ch, int *row, int *col) {
    for (*row = 0; *row < SIZE; (*row)++)
        for (*col = 0; *col < SIZE; (*col)++)
            if (keyTable[*row][*col] == ch)
                return;
}

// Function to perform Playfair encryption or decryption
void playfair(char keyTable[SIZE][SIZE], char message[], char result[]) {
    for (int i = 0, k = 0; i < strlen(message); i += 2) {
        int row1, col1, row2, col2;
        findPosition(keyTable, message[i], &row1, &col1);
        findPosition(keyTable, message[i + 1], &row2, &col2);
                result[k++] = (row1 != row2 && col1 != col2) ? keyTable[row1][col2] :
keyTable[row1][(col1 + 1) % SIZE];
                result[k++] = (row1 != row2 && col1 != col2) ? keyTable[row2][col1] :
keyTable[row2][(col2 + 1) % SIZE];
    }
    result[strlen(message)] = '\0';
}
```

```
int main() {
    char key[25], message[100], result[100];
    char keyTable[SIZE][SIZE];

    // Input the key
    printf("Enter the key: ");
    scanf("%s", key);

    // Input the message
    printf("Enter the message to be encrypted: ");
    scanf("%s", message);

    // Prepare the Playfair Square key table
    prepareKeyTable(key, keyTable);

    // Perform Playfair encryption or decryption
    playfair(keyTable, message, result);

    // Display the encrypted/decrypted message
    printf("Encrypted/Decrypted message: %s\n", result);

    return 0;
}
```

Input and Output:



Result: A c program for demonstration of encrypting and decrypting the messages by Playfair Substitution technique is successfully executed.

SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES