1)C program to implement PLAY FAIR substitution technique

**Program:**

#include <stdio.h>

#include <string.h>

#define N 5 // Size of the grid

// Function to create the grid for the Playfair cipher

void createGrid(char grid[N][N], char\* key)

{

int i, j, k = 0;

for (i = 0; i < N; i++)

{

for (j = 0; j < N; j++)

{

if (k < strlen(key))

grid[i][j] = key[k++];

else

grid[i][j] = 'A' + j + (i \* N);

}

}

}

// Function to encrypt the message using the Playfair cipher

void encrypt(char\* message, char\* key)

{

char grid[N][N];

createGrid(grid, key);

int i, j, k, m, n;

int len = strlen(message);

char newMessage[len];

k = 0;

for (i = 0; i < len; i += 2)

{

m = n = -1;

// Find the position of the first character in the grid

for (j = 0; j < N; j++)

{

for (k = 0; k < N; k++)

{

if (grid[j][k] == message[i])

{

m = j;

n = k;

break;

}

}

if (m != -1 && n != -1)

break;

}

// Find the position of the second character in the grid

for (j = 0; j < N; j++)

{

for (k = 0; k < N; k++)

{

if (grid[j][k] == message[i + 1])

{

m = j;

n = k;

break;

}

}

if (m != -1 && n != -1)

break;

}

if (m == n)

{

newMessage[i] = grid[m][(n + 1) % N];

newMessage[i + 1] = grid[m][(n + 1) % N];

}

else if (m != -1 && n != -1)

{

newMessage[i] = grid[m][n];

newMessage[i + 1] = grid[m][n];

}

}

printf("Encrypted message: %s\n", newMessage);

}

int main()

{

char message[100];

char key[100];

printf("Enter the message to encrypt: ");

scanf("%s", message);

printf("Enter the key: ");

scanf("%s", key);

encrypt(message, key);

return 0;

}

