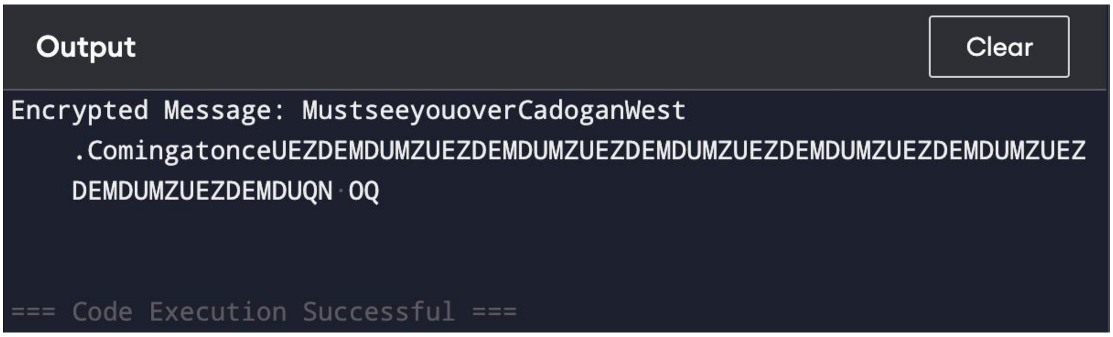
EXP-10:

PROGRAM:

```
#include <stdio.h>
#include <string.h>
#define SIZE 5
char matrix[SIZE][SIZE] = {
   {'M', 'F', 'H', 'I', 'K'},
   {'U', 'N', 'O', 'P', 'Q'},
   {'Z', 'V', 'W', 'X', 'Y'},
   {'E', 'L', 'A', 'R', 'G'},
   {'D', 'S', 'T', 'B', 'C'}
};
void encrypt(char *message) {
   char encrypted[100] = "";
   int i, j, k;
   for (i = 0; message[i] != '\0'; i++) {
      if (message[i] == ' ') continue;
      if (message[i] == 'I' || message[i] == 'J') {
         strcat(encrypted, "I");
      } else {
         strncat(encrypted, &message[i], 1);
      }
   }
   for (i = 0; i < strlen(encrypted); i += 2) {
      if (i + 1 < strlen(encrypted)) {
         char a = encrypted[i];
         char b = encrypted[i + 1];
         int rowA, colA, rowB, colB;
         for (rowA = 0; rowA < SIZE; rowA++) {
            for (colA = 0; colA < SIZE; colA++) {
               if (matrix[rowA][colA] == a) break;
            }
            if (colA < SIZE) break;
         }
         for (rowB = 0; rowB < SIZE; rowB++) {
            for (colB = 0; colB < SIZE; colB++) {
               if (matrix[rowB][colB] == b) break;
            }
            if (colB < SIZE) break;
         }
         if (rowA == rowB) {
            strcat(encrypted, (char[]){matrix[rowA][(colA + 1) % SIZE], '\0'});
            strcat(encrypted, (char[]){matrix[rowB][(colB + 1) % SIZE], '\0'});
         } else if (colA == colB) {
            strcat(encrypted, (char[]){matrix[(rowA + 1) % SIZE][colA], '\0'});
            strcat(encrypted, (char[]){matrix[(rowB + 1) % SIZE][colB], '\0'});
         } else {
            strcat(encrypted, (char[]){matrix[rowA][colB], '\0'});
            strcat(encrypted, (char[]){matrix[rowB][colA], '\0'});
         }
      }
   }
   printf("Encrypted Message: %s\n", encrypted);
}
int main() {
   char message[] = "Must see you over Cadogan West. Coming at once";
   encrypt(message);
   return 0;
}
```

OUTPUT:



```
Encrypted Message: MustseeyouoverCadoganWest
    .ComingatonceUEZDEMDUMZUEZDEMDUMZUEZDEMDUMZUEZDEMDUMZUEZDEMDUMZUEZ
    DEMDUMZUEZDEMDUQN·OQ


=== Code Execution Successful ===
```