PRACTICAL NO 1

Name: Saloni Vishwakarma
Roll no:13
Batch:C1

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void encryptMsg(char msg[], int key)
{
int msgLen = strlen(msg), i, j, k = -1, row = 0, col = 0;
char railMatrix[key][msgLen];
for (i = 0; i < key; ++i)
for (j = 0; j < msgLen; ++j)
railMatrix[i][j] = '\n';
for (i = 0; i < msgLen; ++i)
{
railMatrix[row][col++] = msg[i];
if (row == 0 || row == key - 1)
k = k * (-1);
row = row + k;
}
printf("\nEncrypted Message: ");
for (i = 0; i < key; ++i)
for (j = 0; j < msgLen; ++j)
if (railMatrix[i][j] != '\n')
printf("%c", railMatrix[i][j]);
}
void decryptMsg(char enMsg[], int key)
{
int msgLen = strlen(enMsg), i, j, k = -1, row = 0, col = 0, m = 0;
char railMatrix[key][msgLen];
```

```
for (i = 0; i < key; ++i)
for (j = 0; j < msgLen; ++j)
railMatrix[i][j] = '\n';
for (i = 0; i < msgLen; ++i)
{
railMatrix[row][col++] = '*';
if (row == 0 || row == key - 1)
k = k * (-1);
row = row + k;
}
for (i = 0; i < key; ++i)
for (j = 0; j < msgLen; ++j)
if (railMatrix[i][j] == '*')
railMatrix[i][j] = enMsg[m++];
row = col = 0;
k = -1;
printf("\n\nDecrypted Message: ");
for (i = 0; i < msgLen; ++i)
{
printf("%c", enMsg[i]);
if (row == 0 || row == key - 1)
k = k * (-1);
row = row + k;
}
}
int main()
{
char pt[100], str[100], cp[100], keyt[5][5], abc[25] =
"abcdefghiklmnopqrstuvwxyz";
int i = 0, j = 0, k = 0, n;
printf("Enter plain text: ");
scanf("%s", pt);
fflush(stdin);
printf("Enter key: ");
scanf("%s", str);
```

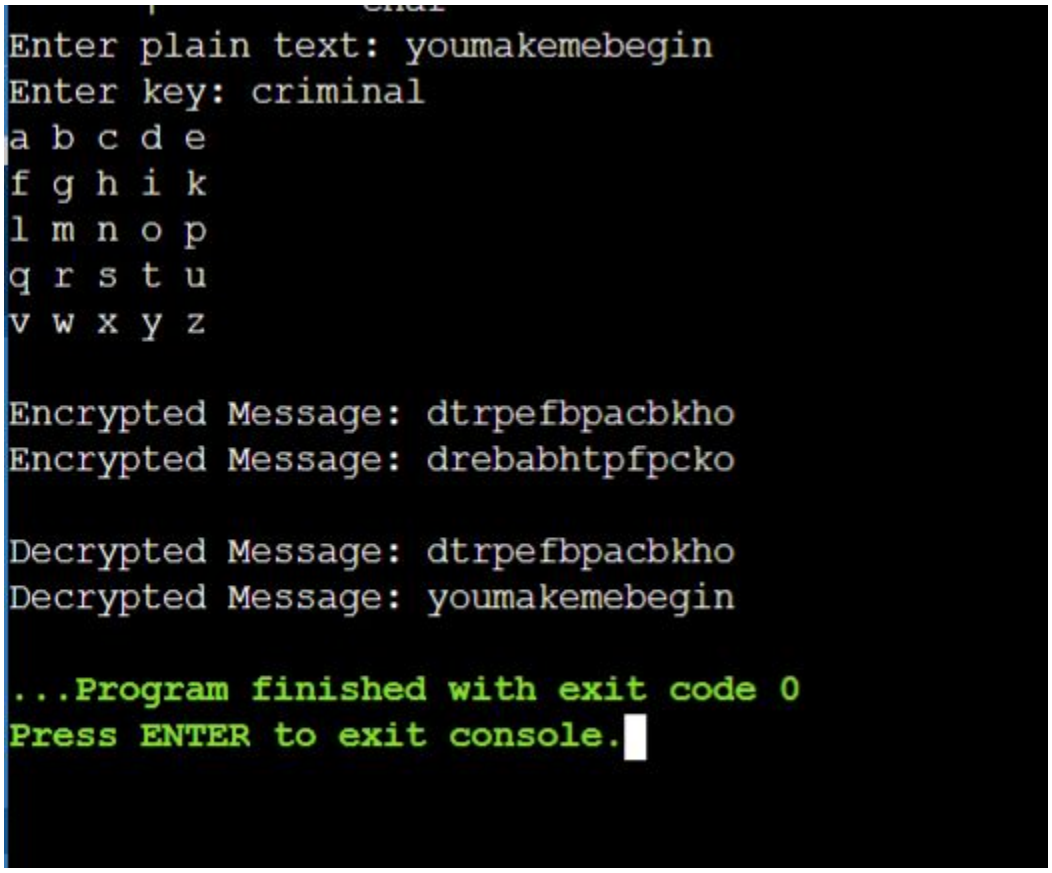
```
strcat(str, abc);
for (i = 0; i < strlen(str); i++)
{
for (j = i + 1; str[j] != '\0'; j++)
{
if (str[j] == str[i])
{
for (k = j; str[k] != '\0'; k++)
{
str[k] = str[k + 1];
}
}
}
}
k = 0;
for (i = 0; i < 5; i++)
{
for (j = 0; j < 5; j++)
{
keyt[i][j] = str[k];
k++;
}
}
for (i = 0; i < 5; i++)
{
for (j = 0; j < 5; j++)
{
printf("%c ", keyt[i][j]);
}
printf("\n");
}
int j1, j2, k1, k2;
for (i = 0; i < strlen(pt); i += 2)
{
for (j = 0; j < 5; j++)
```

```
{
for (k = 0; k < 5; k++)
{
if (pt[i] == keyt[j][k])
{
j1 = j;
k1 = k;
}
if (pt[i + 1] == keyt[j][k])
{
j2 = j;
k2 = k;
}
}
}
if (k1 == k2)
{
j1++;
j2++;
if (j1 == 5)
j1 = 0;
if (j2 == 5)
j2 = 0;
cp[i] = keyt[j1][k1];
cp[i + 1] = keyt[j2][k1];
}
else if (j1 == j2)
{
k1++;
k2++;
if (k1 == 5)
k1 = 0;
if (k2 == 5)
k2 = 0;
cp[i] = keyt[j1][k1];
```

```
cp[i + 1] = keyt[j1][k2];
}
else
{
cp[i] = keyt[j1][k2];
cp[i + 1] = keyt[j2][k1];
}
}
printf("\nEncrypted Message: ");
printf("%s", cp);
int key = 2;
encryptMsg(cp, key);
decryptMsg(cp, key);
for (i = 0; i < strlen(pt); i += 2)
{
for (j = 0; j < 5; j++)
{
for (k = 0; k < 5; k++)
{
if (pt[i] == keyt[j][k])
{
j1 = j;
k1 = k;
}
if (pt[i + 1] == keyt[j][k])
{
j2 = j;
k2 = k;
}
}
}
if (k1 == k2)
{
j1++;
j2++;
```

```
if (j1 == 5)
j1 = 0;
if (j2 == 5)
j2 = 0;
cp[i] = keyt[j1][k1];
cp[i + 1] = keyt[j2][k1];
}
else if (j1 == j2)
{
k1++;
k2++;
if (k1 == 5)
k1 = 0;
if (k2 == 5)
k2 = 0;
cp[i] = keyt[j1][k1];
cp[i + 1] = keyt[j1][k2];
}
else
{
cp[i] = keyt[j1][k2];
cp[i + 1] = keyt[j2][k1];
}
}
printf("\nDecrypted Message: ");
printf("%s", pt);
return 0;
}
```

OUTPUT:


Enter plain text: youmakemebegin
Enter key: criminal
a b c d e
f g h i k
l m n o p
q r s t u
v w x y z
Encrypted Message: dtrpefbpacbkho
Encrypted Message: drebabhtpfpcko
Decrypted Message: dtrpefbpacbkho
Decrypted Message: youmakemebegin
...Program finished with exit code 0
Press ENTER to exit console.