

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef struct{
4      int row;
5      int col;
6
7  }position;
8  char mat[5][5];
9  void generateMatrix(string key)
10 {
11
12     int flag[26]={0};
13     int x =0,y=0;
14     for(int i=0;i<key.length();i++)
15     {
16         if(key[i]=='j')
17             key[i]='i';
18         if(flag[key[i]-'a'] == 0)
19         {
20             mat[x][y++] = key[i];
21             flag[key[i]-'a']=1;
22
23         }
24         if(y==5)x++,y=0;
25     }
26     for(char ch='a';ch<='z';ch++)
27     {
28         if(ch=='j')
29             continue;
30         if(flag[ch-'a']==0)
31         {
32             mat[x][y++]=ch;
33             flag[ch-'a']=1;
34         }
35         if(y==5)x++,y=0;
36     }
37 }
38 string formatMessage(string msg)
39 {
40
41     for(int i=0;i<msg.length();i++)
42     {
43         if(msg[i]=='j')msg[i]='i';
44     }
45     for(int i=1;i<msg.length();i+=2)
46     {
47
48         if(msg[i-1] ==msg[i])msg.insert(i, "x");
49
50     }
51
52     if(msg.length()%2!=0)msg += "x";
53     return msg;
54 }
55 position getPosition(char c)
56 {
57     for(int i=0;i<5;i++)
58         for(int j =0;j<5;j++)
59             if(c == mat[i][j])
60             {
61
62                 position p = {i,j};
63                 return p;
64             }
65 }
66 string encrypt(string message)

```

```

67 {
68
69     string ctext="";
70     for(int i =0;i<message.length();i+=2)
71     {
72
73         position p1 = getposition(message[i]);
74         position p2 = getposition(message[i+1]);
75         int x1 = p1.row;
76         int y1=p1.col;
77         int x2=p2.row;
78         int y2=p2.col;
79
80         if( x1 == x2 ) // same row
81         {
82             ctext.append(1,mat[x1][(y1+1)%5]);
83             ctext.append(1,mat[x2][(y2+1)%5]);
84         }
85         else if( y1 == y2 ) // same column
86         {
87             ctext.append(1,mat[ (x1+1)%5 ][ y1 ]);
88             ctext.append(1,mat[ (x2+1)%5 ][ y2 ]);
89         }
90         else
91         {
92             ctext.append(1,mat[ x1 ][ y2 ]);
93             ctext.append(1,mat[ x2 ][ y1 ]);
94         }
95     }
96 }
97 return ctext;
98 }
99 string Decrypt(string message)
100 {
101
102     string ptext;
103     for(int i=0;i<message.length();i+=2)
104     {
105
106         position p1=getposition(message[i]);
107         position p2=getposition(message[i+1]);
108         int x1=p1.row;int y1=p1.col;
109         int x2=p2.row;int y2=p2.col;
110         if( x1 == x2 ) // same row
111         {
112             ptext.append(1,mat[x1][ --y1<0 ? 4: y1 ]);
113             ptext.append(1,mat[x2][ --y2<0 ? 4: y2 ]);
114         }
115         else if( y1 == y2 ) // same column
116         {
117             ptext.append(1,mat[ --x1<0 ? 4: x1 ][y1]);
118             ptext.append(1,mat[ --x2<0 ? 4: x2 ][y2]);
119         }
120         else
121         {
122             ptext.append(1,mat[ x1 ][ y2 ]);
123             ptext.append(1, mat[ x2 ][ y1 ]);
124         }
125     }
126     return ptext;
127 }
128 int main()
129 {
130     string plaintext;
131     cout<<"enter message:";cin>>plaintext;
132     int n;

```

```

133     cout<<"enter th no of keys:";cin>>n;
134     string key[n];
135     for(int i=0;i<n;i++)
136     {
137         cout << "enter no of keys :" << i+1<<":"<<key[i];
138         cin>>key[i];
139         generateMatrix(key[i]);
140         cout<<"key"<<i+1<<"matrix:"<<endl;
141         for(int k=0;k<5;k++)
142         {
143             for(int j=0;j<5;j++)
144             {
145
146                 cout<<mat[k][j]<<" ";
147
148             }
149             cout<<endl;
150
151         }
152         cout<<"actual message\t\t"<<plaintext<<endl;
153         string fmsg=formatMessage(plaintext);
154         cout<<"formated message\t"<<fmsg<<endl;
155
156         string ciphertext = encrypt(fmsg);
157         cout<<"encrypted message\t"<<ciphertext<<endl;
158         string decryptmsg =Decrypt(ciphertext);
159         cout<<"Decrypted message\t:"<<decryptmsg<<endl;
160     }
161 }
162
163

```