PROGRAM:

```
#include<iostream>
#include <bits/stdc++.h>

using namespace std;
int keyMatrix[100][100], inverseMatrix[100][100];
int order;

string FormatMessage(string message)
{
   for(int i=0;i<message.length();i++)
   {
      if(message[i] == ' ')
         message = message.replace(i, 1, "");
      if(message[i] == 'j')
         message = message.replace(i, 1, "i");
   }
   for(int i=1;i<message.length();i++)
      if(message[i-1] == message[i])
         message = message.insert(i, "x"), i++;
      if(message.length()%2)
         message += "x";
   return message;
}
int GetInverseDeterminant(int R , int D = 26)
{
   int i =0 ;
   int p0= 0 , p1 =1 ;
   int q = 1 ;
   int q0 , q1 ;
   while(R!=0)
   {
      q = D/R ;
      int tempD = D ;
      D=R;
      R = tempD%R ;

      if(i==0)
      {
         p0 = 0 ;
         q0 = q ;
      }

      else if(i==1)
      {
         p1=1 ;
         q1=q ;
      }

      else
      {
```

```
            int temp = p1 ;
            p1 = (p0-p1*(q0))%26 ;
            if(p1<0)
               p1 = 26-(abs(p1)%26) ;
            p0 = temp ;
            q0 = q1;
            q1 = q ;
         }

         i++ ;
      }
      p1 = (p0-p1*(q0))%26 ;
      return p1 ;
   }
   int GetDeterminant()
   {
      int determinant = 0;
      if(order==2)
         determinant = keyMatrix[0][0] * keyMatrix[1][1] - keyMatrix[0][1] * keyMatrix[1][0];
      else
         for(int i = 0; i < 3; i++)
            determinant = determinant + (keyMatrix[0][i] * (keyMatrix[1][(i+1)%3] *
      keyMatrix[2][(i+2)%3] - keyMatrix[1][(i+2)%3] * keyMatrix[2][(i+1)%3]));

      if(determinant<0)
         determinant = 26 - (int(-determinant)%26);
      else
         determinant = int(determinant)%26;
      determinant = GetInverseDeterminant(determinant, 26);
      return determinant;
   }
   string Multiply(string msg_group, int matrix[][100])
   {
      string result;
      for(int i=0; i<order; i++)
      {
         float val = 0 ;
         for(int j=0;j<order; j++)
            val = val + matrix[j][i] * (msg_group[j] - 'a');
         if(val>=0)
            val = int(val)%26 + 'a';
         else
            val = 26 - (int(-val)%26) + 'a';
         result += int(val);
      }
      return result;
   }
   void FindInverse(int determinant)
   {
      if(order==2)
      {
         inverseMatrix[0][0] = keyMatrix[1][1]*determinant;
```

```
        inverseMatrix[1][1] = keyMatrix[0][0]*determinant;
        inverseMatrix[0][1] = -keyMatrix[0][1]*determinant;
        inverseMatrix[1][0] = -keyMatrix[1][0]*determinant;
    }
    else
        for(int j=0; j<order; j++)
        for(int i=0; i<order; i++)
        inverseMatrix[i][j] = ((keyMatrix[(j+1)%3][(i+1)%3] * keyMatrix[(j+2)%3][(i+2)%3]) -
(keyMatrix[(j+1)%3][(i+2)%3] * keyMatrix[(j+2)%3][(i+1)%3]))*determinant;
    for(int j=0; j<order; j++)
        for(int i=0; i<order; i++)
        if(inverseMatrix[i][j] < 0 ) inverseMatrix[i][j] = 26 - int(-inverseMatrix[i][j])%26;
    else inverseMatrix[i][j] = int(inverseMatrix[i][j]) %26;
}
string Encrypt(string message)
{
    string enc_msg;
    for(int i=0;i<message.length();i+=order)
    {
        string msg_group = message.substr(i, order);
        msg_group = Multiply(msg_group, keyMatrix);
        enc_msg = enc_msg.append(msg_group);
    }
    return enc_msg;
}
string Decrypt(string message)
{
    string msg;
    FindInverse(GetDeterminant());
    for(int i=0;i<message.length();i+=order)
    {
        string msg_group = message.substr(i, order);
        msg_group = Multiply(msg_group, inverseMatrix);
        msg = msg.append(msg_group);
    }
    return msg;
}
int main()
{
    string message;
    cout << "Enter the key matrix order: ";
    cin >> order;
    cout << "Enter the keys:";
    for(int i = 0; i < order; i++)
        for(int j=0;j<order; j++)
            cin >> keyMatrix[i][j];
    cin.get();
    cout<<"Enter a message to be encrypted: ";
    getline(cin, message);
    FormatMessage(message);
    string enc_msg = Encrypt(message);
    string dec_msg = Decrypt(enc_msg);
```

```
    cout<<"Message: "<<message<<endl;
    cout<<"Encrypted Message: "<<enc_msg<<endl;
    cout<<"Decrypted Message: "<<dec_msg<<endl;
    return 0;
}
```

OUTPUT:

1:
```
Enter the key matrix order: 2
Enter the keys:
3       3
2       5
Enter a message to be encrypted: hillcipher

Message: hillcipher

Encrypted Message: ljdkwuhcut

Decrypted Message: hillcipher
```

2:
```
Enter the key matrix order: 3
Enter the keys:
17      17      5
21      18      21
2       2       19
Enter a message to be encrypted: paymoremoney

Message: paymoremoney

Encrypted Message: rrlmwbkaspdh

Decrypted Message: paymoremoney
```