```cpp
#include <bits/stdc++.h>
using namespace std;
typedef struct{
    int row;
    int col;

}position;
char mat[5][5];
void generateMatrix(string key)
{

    int flag[26]={0};
    int x =0,y=0;
    for(int i=0;i<key.length();i++)
    {
        if(key[i]=='j')
            key[i]='i';
        if(flag[key[i]-'a'] == 0)
        {
            mat[x][y++] = key[i];
            flag[key[i]-'a']=1;

        }
        if(y==5)x++,y=0;
    }
for(char ch='a';ch<='z';ch++)
{
    if(ch=='j')
        continue;
    if(flag[ch-'a']==0)
    {
        mat[x][y++]=ch;
        flag[ch-'a']=1;
    }
    if(y==5)x++,y=0;
}
}
string formatMessage(string msg)
{

    for(int i=0;i<msg.length();i++)
    {
        if(msg[i]=='j')msg[i]='i';
    }
        for(int i=1;i<msg.length();i+=2)
        {

            if(msg[i-1] ==msg[i])msg.insert(i,"x");

        }

        if(msg.length()%2!=0)msg +="x";
        return msg;
}
position getposition(char c)
{
    for(int i=0;i<5;i++)
        for(int j =0;j<5;j++)
        if(c == mat[i][j])
    {

        position p = {i,j};
        return p;
    }
}
string encrypt(string message)
```

```
{

    string ctext="";
    for(int i =0;i<message.length();i+=2)
    {

        position p1 = getposition(message[i]);
        position p2 = getposition(message[i+1]);
        int x1 = p1.row;
        int y1=p1.col;
        int x2=p2.row;
        int y2=p2.col;

     if( x1 == x2 ) // same row
        {
            ctext.append(1,mat[x1][(y1+1)%5]);
            ctext.append(1,mat[x2][(y2+1)%5]);
        }
        else if( y1 == y2 ) // same column
        {
            ctext.append(1,mat[ (x1+1)%5 ][ y1 ]);
            ctext.append(1,mat[ (x2+1)%5 ][ y2 ]);
        }
        else
        {
            ctext.append(1,mat[ x1 ][ y2 ]);
            ctext.append(1,mat[ x2 ][ y1 ]);
        }

    }
    return ctext;
}
string Decrypt(string message)
{

    string ptext;
    for(int i=0;i<message.length();i+=2)
    {

        position p1=getposition(message[i]);
        position p2=getposition(message[i+1]);
        int x1=p1.row;int y1=p1.col;
        int x2=p2.row;int y2=p2.col;
        if( x1 == x2 ) // same row
        {
            ptext.append(1,mat[x1][ --y1<0 ? 4: y1 ]);
            ptext.append(1,mat[x2][ --y2<0 ? 4: y2 ]);
        }
        else if( y1 == y2 ) // same column
        {
            ptext.append(1,mat[ --x1<0 ? 4: x1 ][y1]);
            ptext.append(1,mat[ --x2<0 ? 4: x2 ][y2]);
        }
        else
        {
            ptext.append(1,mat[ x1 ][ y2 ]);
            ptext.append(1, mat[ x2 ][ y1 ]);
        }
    }
    return ptext;
}
int main()
{
    string plaintext;
    cout<<"enter message:";cin>>plaintext;
    int n;
```

```
    cout<<"enter th no of keys:";cin>>n;
    string key[n];
    for(int i=0;i<n;i++)
    {
    cout << "enter no of keys :" << i+1<<":"<<key[i];
    cin>>key[i];
    generateMatrix(key[i]);
    cout<<"key"<<i+1<<"matrix:"<<endl;
    for(int k=0;k<5;k++)
    {
        for(int j=0;j<5;j++)
        {

            cout<<mat[k][j]<<" ";

    }
        cout<<endl;

    }
    cout<<"actual message\t\t"<<plaintext<<endl;
    string fmsg=formatMessage(plaintext);
    cout<<"formated message\t"<<fmsg<<endl;

    string ciphertext = encrypt(fmsg);
    cout<<"encrypted message\t"<<ciphertext<<endl;
    string decryptmsg =Decrypt(ciphertext);
    cout<<"Decrypted message\t:"<<decryptmsg<<endl;
}
}
```