

# **INFORMATION SECURITY (3170720)**



**VVP  
ENGINEERING  
COLLEGE**

**SUBMITTED BY: DISHEN MAKWANA**

**180470107035**

**G2**



# V. V. P. Engineering College, Rajkot

## Department of Computer Engineering

---

### **Vision of the Institute**

- To be an exemplary institute, transforming students into competent professionals with human values.

### **Mission of the Institute**

- To provide a conducive academic environment for strengthening technical capabilities of the students.
- To strengthen linkage with industries, alumni and professional bodies.
- To organize various co-curricular and extra-curricular activities for overall development of the students.
- To practice good governance and conduct value- based activities for making students responsible citizens.

### **Vision of the Department**

- Transforming students into globally efficient professionals with moral values.

### **Mission of the Department**

- To provide a strong foundation of computer engineering through effective teaching learning process.
- To enhance industry linkage & alumni network for better placement and real-world exposure.
- To provide various opportunities & platforms for all round development of students &

encourage them for value-based practices.

## **Program Educational Objectives (PEOs)**

Graduates will be able to

- Apply computer engineering theories, principles and skills to meet the challenges of the society.
- Communicate effectively, work collaboratively and manifest professionalism with ethics.
- Exhibit life-long learning attitude and adapt to rapid technological changes in industry.
- Advance their career in industry, pursue higher education or become an entrepreneur.



# V.V.P. ENGINEERING COLLEGE

## RAJKOT

### Certificate

This is to certify that

Mr. DISHEN MAKWANA, Enrollment No: 180470107035, Branch: Computer Engineering, Semester: 7 has satisfactorily completed the course in the subject: **INFORMATION SECURITY (3170720)** within the four walls of V.V.P. Engineering College, Rajkot.

Date of Submission:

---

**Prof. Komil Vora,**  
Staff In-Charge

Head of Department,  
Department of Computer Engineering,  
V.V.P. Engineering College



**V. V. P. Engineering College**  
**Department of Computer Engineering**  
**Course Outcomes**

Semester: 7<sup>th</sup>

Subject: Information Security

Subject Code: 3170720

After learning the course, the students will be able to:

<b>CO Number</b>	<b>Course Outcomes</b>	<b>CL</b>
C3170720.1	Explain the different attacks possible during data transmission and need of cryptography.	U
C3170720.2	Apply various symmetric and asymmetric algorithms.	Ap
C3170720.3	Assess the performance of Hash function, MAC function and Digital signature.	E
C3170720.4	Classify the different techniques of key distribution.	U
C3170720.5	Compare remote user authentication techniques and security protocols.	A

## Index

EXP NO.	EXPERIMENT	LAB	Sign
1	Breaking the shift cipher	1	
2	Breaking the monoalphabetic cipher	2	
3	Vernam Cipher and perfect secrecy	3	
4	Playfair Cipher	4	
5	Hill Cipher	5	

# LAB 1

## • Shift Cipher

### 1. Encrypt the following plaintext using key = 7:

Lord Rama was a good king.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

0123456789 0123456789 012345

L=11 O=14 R=17 D=3 R=17 A=0 M=12 A=0 W=22 A=0 S=18 A=0 G=6 O=14 O=14 D=3 K=10  
I=8 N=13 G=6

11 14 17 3 17 0 12 0 22 0 18 0 6 14 14 3 10 8 13 6

Ans: +7

18 21 24 10 24 7 19 7 29/4 7 25 7 13 21 21 10 17 15 20 13

Ans: Svyk yhth dhz h nvvk rpun

### 2. Given the plain text.

Plaintext: plain text and its corresponding cipher text, find out the key used for the encryption of  
abcdefghijklmnopqrstuvwxyz

Ciphertext: TDNUCBZROHLGYVFPWIXSEKAMQJ

Ans: Error

### 3. How many different keys are possible with an n-letter alphabet?

Ans: 25

### 4. Given a ciphertext, find out the corresponding plain text using brute force attack:

Ciphertext: HAAHJR HA KHDU

Ans: 19 Attack at dawn

**Code :**

```
#include <iostream>
using namespace std;
```

```
string encrypt(string text, int s)
{
    string result = "";
    for (int i=0;i<text.length();i++)
    {
        if (isupper(text[i]))
            result += char(int(text[i]+s-65)%26 +65);

        else
            result += char(int(text[i]+s-97)%26 +97);
    }
    return result;
}
```

```
int main()
```

```

{
    string text="ATTACKATONCE";
    int s = 4;
    cout << "Text : " << text;
    cout << "\nShift: " << s;
    cout << "\nCipher: " << encrypt(text, s);
    return 0;
}

```

```

Shift Cipher > C++ code.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  string encrypt(string text, int s)
5  {
6      string result = "";
7      for (int i = 0; i < text.length(); i++)
8      {
9          if (isupper(text[i]))
10             result += char(int(text[i] + s - 65) % 26 + 65);
11
12         else
13             result += char(int(text[i] + s - 97) % 26 + 97);
14     }
15     return result;
16 }
17
18 int main()
19 {
20     string text = "HAAHJRHA KH DU";
21     int s = 4;
22     cout << "Text : " << text;
23     cout << "\nShift: " << s;
24     cout << "\nCipher: " << encrypt(text, s);
25     return 0;
}

```

PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE

<https://aka.ms/powershell>  
Type 'help' to get help.

A new PowerShell preview release is available: v7.2.0-preview.8  
Upgrade now, or check out the release page at:  
<https://aka.ms/PowerShell-Release?tag=v7.2.0-preview.8>

```

PS D:\INS> cd "d:\INS\Shift Cipher\" ; if ($?) { g++ code.cpp -o code } ; if ($?) { .\code }
Text : HAAHJRHA KH DU
Shift: 4
Cipher: LEELNVLEOLHY
PS D:\INS\Shift Cipher>

```



## LAB 2

### • Monoalphabetic Cipher

Key:

Plain: a b c d e f g h i j k l m n o p q r s t u v w x y z

Cipher: DKVQFIBJWPESCXHTMYAUOLRGZN

Plaintext: if we wish to replace letters

Ciphertext: WI RF RWAJ UH YFTSDVF SFUUFYA

### Quiz:

#### 1) Explain Monoalphabetic cipher.

Monoalphabetic cipher is an improvement over caesar cipher. In this cipher each letter has a defined alphabet and it is assigned to it in every occurrence. Like

QWERTYUIOPASDFGHJKLZXCVBNM key is assigned to  
ABCDEFGHIJKLMNOPQRSTUVWXYZ

So Hello world becomes URAAF VFKA E

#### 2) Justify why the Monoalphabetic cipher is more secure than the Caesar cipher.

In a Caesar cipher the possible number of guessing the key is only 25 but in a monoalphabetic cipher, It's 26!. So to avoid brute force attack, a monoalphabetic cipher is more secure than a caesar cipher.

#### 3) Create your key and convert following sentence in to cipher text

A	Q	I	Z
B	A	W	C
C	Z	D	B
D	W	J	M
E	S	Q	N
F	X	P	V
G	E	E	X
H	D	U	A
I	C	F	D
J	R	H	G
K	F	R	J
L	V	Y	L

M	T	T	K
N	G	A	H
O	B	L	F
P	Y	Z	S
Q	H	C	Q
R	N	X	E
S	U	S	T
T	J	K	U
U	M	M	O
V	I	B	P
W	K	N	I
X	O	V	Y
Y	L	P	R
Z	P	F	W

a. A quick brown fox jump over the lazy dog

ANS: Q HMCZF ANBKG XBO RMTY BISN JDS VQPL WBE

b. I am student of vvp engg college

ANS: F IT SKMJQAK LP BBZ QAEE DLYYQEQ

c. Gandhinagar is capital of gujarat

ANS: XZHMADHZXZE DT BZSDUZL FV XOGZEUZU

**Code:**

```
#include <bits/stdc++.h>
using namespace std;
unordered_map<char,char> hashMap;

string encrypt(string msg)
{
    string ciphertext;
    for(int i=0; i<msg.size(); i++)
    {
        ciphertext.push_back(hashMap[msg[i]]);
    }

    return ciphertext;
}
```

```

}

string decrypt(string msg)
{
    string plaintext;
    for(int i=0; i<msg.size(); i++)
    {
        plaintext.push_back(hashMap[msg[i]]);
    }

    return plaintext;
}

void hashFn(string a, string b)
{
    hashMap.clear();
    for(int i=0; i<a.size(); i++)
    {
        hashMap.insert(make_pair(a[i],b[i]));
    }
}

int main()
{
    string alphabet = "abcdefghijklmnopqrstuvwxyz";
    string substitution = "qwertyuiopasdfghjklzxcvbnm";
    string msg = "absdhj";

    hashFn(alphabet, substitution);

    string cipher = encrypt(msg);
    cout<<"Encrypted Cipher Text: "<<cipher<<endl;

    hashFn(substitution, alphabet);
    string plain = decrypt(cipher);
    cout<<"Decrypted Plain Text: "<<plain<<endl;
}

```

C++ Monoalphabetic\_Cipher.cpp X

C++ Monoalphabetic\_Cipher.cpp > hashFn(string, string)

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  unordered_map<char, char> hashMap;
4
5  string encrypt(string msg)
6  {
7      string ciphertext;
8      for (int i = 0; i < msg.size(); i++)
9      {
10         ciphertext.push_back(hashMap[msg[i]]);
11     }
12     return ciphertext;
13 }
14
15 string decrypt(string msg)
16 {
17     string plaintext;
18     for (int i = 0; i < msg.size(); i++)
19     {
20         plaintext.push_back(hashMap[msg[i]]);
21     }
22 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

A new PowerShell preview release is available: v7.2.0-preview.8  
Upgrade now, or check out the release page at:  
<https://aka.ms/PowerShell-Release?tag=v7.2.0-preview.8>

```
PS D:\INS> cd "d:\INS\Shift Cipher" ; if ($?) { g++ code.cpp -o code } ; if ($?) { .\code }
Text : HAAHJRHAKHDU
Shift: 4
Cipher: LEELNVLEOLHY
PS D:\INS\Shift Cipher> cd "d:\INS\Monoalphabetic Cipher" ; if ($?) { g++ code.cpp -o code }
Encrypted Cipher Text: qwlrip
Decrypted Plain Text: absdhj
PS D:\INS\Monoalphabetic Cipher>
```

## LAB 3

- Vernam Cipher

### 1) Execute below vernam cipher

**Code:**

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int t,n,i,j,k,sum=0;
    string m;
    cout<<"Enter the message"<<"\n";
    cin>>m;
    string key;
    cout<<"Enter the key"<<"\n";
    cin>>key;
    int mod = key.size();
    j=0;
    for(i=key.size();i<m.size();i++){
        key+=key[j%mod];
        j++;
    }
    string ans="";
    for(i=0;i<m.size();i++){
        ans += (key[i]-'A'+m[i]-'A')%26+'A';
    }
    cout<<"Encrypted message: "<<ans<<"\n";

    return 0;
}
```

The screenshot shows a C++ IDE with two tabs: 'Monoalphabetic\_Cipher.cpp' and 'code.cpp'. The 'code.cpp' tab is active, displaying the following code:

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main()
5  {
6      int t, n, i, j, k, sum = 0;
7      string m;
8      cout << "Enter the message" << '\n';
9      cin >> m;
10     string key;
11     cout << "Enter the key" << '\n';
12     cin >> key;
13     int mod = key.size();
14     j = 0;
15     for (i = key.size(); i < m.size(); i++)
16     {
17         key += key[j % mod];
18         j++;
19     }
20     string ans = "";
21     for (i = 0; i < m.size(); i++)
22     {
23         ans += (key[i] - 'A' + m[i] - 'A') % 26 + 'A';
24     }
25     cout << "Encrypted message: " << ans << '\n';

```

Below the code, the 'TERMINAL' tab is active, showing the execution of the program:

```

Shift: 4
Cipher: LEELNVLEOLHY
PS D:\INS\Shift Cipher> cd "d:\INS\Monoalphabetic Cipher\" ; if ($?) { g++ code.cpp -o code } ; if ($?)
Encrypted Cipher Text: qwlrip
Decrypted Plain Text: absdhj
PS D:\INS\Monoalphabetic Cipher> cd "d:\INS\Vernam Cipher\" ; if ($?) { g++ code.cpp -o code } ; if ($?)
Enter the message
vernampcipher
Enter the key
kjendathoper
Encrypted message: RZHMPYHBPIUU
PS D:\INS\Vernam Cipher>

```

**EX:**

Plain Text : vernampcipher

Key : kjendathoper

Ans : RZHMPYHBPIUU

## 2) Decrypt below vernam cipher

**Code:**

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

```
using namespace std;
```

```

int main(){
    int t,n,i,j,k,sum=0;
    string m;
    cout<<"Enter the message"<<"\n";
    cin>>m;
    string key;
    cout<<"Enter the key"<<"\n";
    cin>>key;
    int mod = key.size();
    j=0;
    for(i=key.size();i<m.size();i++){

```

```

        key+=key[j%mod];
        j++;
    }
    string ans="";
    for(i=0;i<m.size();i++){
        ans += (m[i]-key[i]+26)%26+'A';
    }
    cout<<"Decrypted message: "<<ans<<"\n";

    return 0;
}

```

The screenshot shows a C++ IDE with three tabs: Monoalphabetic\_Cipher.cpp, code.cpp, and de.cpp. The active tab is de.cpp, which contains the following code:

```

Vernam Cipher > C++ de.cpp > main()
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main()
5  {
6      int t, n, i, j, k, sum = 0;
7      string m;
8      cout << "Enter the message" << '\n';
9      cin >> m;
10     string key;
11     cout << "Enter the key" << '\n';
12     cin >> key;
13     int mod = key.size();
14     j = 0;
15     for (i = key.size(); i < m.size(); i++)
16     {
17         key += key[j % mod];
18         j++;
19     }
20     string ans = "";
21     for (i = 0; i < m.size(); i++)
22     {
23         ans += (m[i] - key[i] + 26) % 26 + 'A';
24     }
25     cout << "Decrypted message: " << ans << '\n';

```

The terminal output shows the execution of the program:

```

Enter the message
vernamcipher
Enter the key
kjendathoper
Encrypted message: RZHPYHBPIUU
PS D:\INS\Vernam Cipher> cd "d:\INS\Vernam Cipher\" ; if ($?) { g++ de.cpp -o de } ; if ($?) { .\de }
Enter the message
wixksdbeswpthmezkwcusqp
Enter the key
tevtuoiwejhbuylkwekmrfl
Decrypted message: DECRYPTIONISNOTPOSSIBLE
PS D:\INS\Vernam Cipher>

```

**EX:**

Cipher Text : wixksdbeswpthmezkwcusqp

Key : tevtuoiwejhbuylkwekmrfl

Ans : DECRYPTIONISNOTPOSSIBLE

## LAB 4

### • Playfair Cipher

Plain Text : jazz

Key : monarchy

I/J ARE TOGETHER

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Divide plain texts to two pairs:

jazz => ja zx zx

greet => gr ex et

off => of fx

ja => sb

In one column then immediate bottom

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

xz => zu

In case of one row take immediate right

All rows and columns are roundly connected.



M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

sz:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

ru:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

nt:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

1) Encrypt following:

**Code:**

```
#include<iostream>
#include<string>
#include<vector>
#include<map>
using namespace std;
int main(){
    int i,j,k,n;
    cout<<"Enter the message"<<endl;
    string s,origin;
    getline(cin,origin);
    cout<<"Enter the key"<<endl;
    string key;
    cin>>key;
    for(i=0;i<origin.size();i++){
        if(origin[i]!=' '){
            s+= origin[i];
        }
    }
    vector<vector<char>> > a(5,vector<char>(5,' '));
```

```

n=5;
map<char,int> mp;
k=0;
int pi,pj;
for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        while(mp[key[k]]>0&& k<key.size()){
            k++;
        }
        if(k<key.size()){
            a[i][j]=key[k];
            mp[key[k]]++;
            pi=i;
            pj=j;
        }
        if(k==key.size())
            break;
    }
    if(k==key.size())
        break;
}
k=0;
for(;i<n;i++){
    for(;j<n;j++){
        while(mp[char(k+'a')]>0&& k<26){
            k++;
        }
        if(char(k+'a')=='j'){
            j--;
            k++;
            continue;
        }
        if(k<26){
            a[i][j]=char(k+'a');
            mp[char(k+'a')]++;
        }
    }
    j=0;
}

```

```

string ans;
if(s.size()%2==1)
    s+="x";
for(i=0;i<s.size()-1;i++){
    if(s[i]==s[i+1])
        s[i+1]='x';
}

```

```

map<char,pair<int,int> > mp2;

```

```

for(i=0;i<n;i++){

```

```

        for(j=0;j<n;j++){
            mp2[a[i][j]] = make_pair(i,j);
        }
    }

    for(i=0;i<s.size()-1;i+=2){
        int y1 = mp2[s[i]].first;
        int x1 = mp2[s[i]].second;
        int y2 = mp2[s[i+1]].first;
        int x2 = mp2[s[i+1]].second;
        if(y1==y2){
            ans+=a[y1][(x1+1)%5];
            ans+=a[y1][(x2+1)%5];
        }
        else if(x1==x2){
            ans+=a[(y1+1)%5][x1];
            ans+=a[(y2+1)%5][x2];
        }
        else {
            ans+=a[y1][x2];
            ans+=a[y2][x1];
        }
    }
    cout<<ans<<'\n';
    return 0;
}

```

```

C++ en.cpp
playfair cipher > C++ en.cpp > main()
175     encrypt(str, keyT, ps);
176 }
177
178 // Driver code
179 int main()
180 {
181     char str[SIZE], key[SIZE];
182
183     // Key to be encrypted
184     strcpy(key, "playfair");
185     printf("key text: %s\n", key);
186
187     // Plaintext to be encrypted
188     strcpy(str, "instruments ");
189     printf("Plain text: %s\n", str);
190
191     // encrypt using Playfair Cipher
192     encryptByPlayfairCipher(str, key);
193
194     printf("Cipher text: %s\n", str);
195
196     return 0;
197 }
198
PROBLEMS 4 OUTPUT TERMINAL DEBUG CONSOLE
Encrypted message: RZHPYHBPIUU
PS D:\INS\Vernam Cipher> cd "d:\INS\Vernam Cipher\" ; if ($?) { g++ de.cpp -o de } ; if ($?) { .\de }
Enter the message
wixksdbeswpthmezkwcsqp
Enter the key
tevtuoiewjhbuylkwekmrfl
Decrypted message: DECRYPTIONISNOTPOSSIBLE
PS D:\INS\Vernam Cipher> cd "d:\INS\playfair cipher\" ; if ($?) { g++ en.cpp -o en } ; if ($?) { .\en }
Key text: playfair
Plain text: instruments
Cipher text: eutniveagontx
PS D:\INS\playfair cipher>

```

Message : instruments

Key : playfair

Ans : eutnivegontx

2. Decrypt following:

**Code:**

```
#include<iostream>
#include<string>
#include<vector>
#include<map>
using namespace std;
int main(){
    int i,j,k,n;
    cout<<"Enter the encrypted message\n";
    string s;
    cin>>s;
    cout<<"Enter the key\n";
    string key;
    cin>>key;
    vector<vector<char>> > a(5,vector<char>(5,' '));
    n=5;
    map<char,int> mp;
    k=0;
    int pi,pj;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            while(mp[key[k]]>0&& k<key.size()){
                k++;
            }
            if(k<key.size()){
                a[i][j]=key[k];
                mp[key[k]]++;
                pi=i;
                pj=j;
            }
            if(k==key.size())
                break;
        }
        if(k==key.size())
            break;
    }
    k=0;
    for(;i<n;i++){
        for(;j<n;j++){
            while(mp[char(k+'a')]>0&& k<26){
                k++;
            }
            if(char(k+'a')== 'j'){
                j--;
                k++;
            }
        }
    }
}
```

```

        continue;
    }
    if(k<26){
        a[i][j]=char(k+'a');
        mp[char(k+'a')]++;
    }
}
j=0;
}

string ans;

map<char,pair<int,int> > mp2;

for(i=0;i<n;i++){
    for(j=0;j<n;j++){
        mp2[a[i][j]] = make_pair(i,j);
    }
}
for(i=0;i<s.size()-1;i+=2){
    int y1 = mp2[s[i]].first;
    int x1 = mp2[s[i]].second;
    int y2 = mp2[s[i+1]].first;
    int x2 = mp2[s[i+1]].second;
    if(y1==y2){
        ans+=a[y1][(x1-1)%5];
        ans+=a[y1][(x2-1)%5];
    }
    else if(x1==x2){
        ans+=a[(y1-1)%5][x1];
        ans+=a[(y2-1)%5][x2];
    }
    else {
        ans+=a[y1][x2];
        ans+=a[y2][x1];
    }
}
if(ans[ans.size()-1]=='x')
    ans[ans.size()-1]='\0';

for(i=1;i<ans.size();i++){
    if(ans[i]=='x')
        ans[i]=ans[i-1];
}

cout<<ans<<'\n';
return 0;
}

```

```
en.cpp de.cpp x
playfair cipher > C++ de.cpp > main()
158
159 // Driver code
160 int main()
161 {
162     char str[SIZE], key[SIZE];
163
164     // Key to be encrypted
165     strcpy(key, "Gravity Falls");
166     printf("Key text: %s\n", key);
167
168     // Ciphertext to be decrypted
169     strcpy(str, "GFFGBMGFNFAW");
170     printf("Plain text: %s\n", str);
171
172     // encrypt using Playfair Cipher
173     decryptByPlayfairCipher(str, key);
174
175     printf("Deciphered text: %s\n", str);
176
177     return 0;
178 }
179
180 // This code is contributed by AbhayBhat
181
```

PROBLEMS 11 OUTPUT TERMINAL DEBUG CONSOLE

```
Enter the key
tevtuoiwejhbuylkwekmrfl
Decrypted message: DECRYPTIONISNOTPOSSIBLE
PS D:\INS\Vernam Cipher> cd "d:\INS\playfair cipher\" ; if ($?) { g++ en.cpp -o en } ; if ($?) { .\en }
Key text: playfair
Plain text: instruments
Cipher text: eutnivegontx
PS D:\INS\playfair cipher> cd "d:\INS\playfair cipher\" ; if ($?) { g++ de.cpp -o de } ; if ($?) { .\de }
Key text: Gravity Falls
Plain text: GFFGBMGFNFAW
Deciphered text: attackatdawn
PS D:\INS\playfair cipher>
```

**EX:**

Cipher Text : GFFGBMGFNFAW

Key : Gravity Falls

Ans : attackatdawn

## LAB 5

- **Hill Cipher**

1. Encrypt Following using Hill Cipher

**Code:**

```
#include<iostream>
#include<vector>
using namespace std;
int main(){
    int x,y,i,j,k,n;
    cout<<"Enter the size of key matrix\n";
    cin>>n;
    cout<<"Enter the key matrix\n";
    int a[n][n];
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            cin>>a[i][j];
        }
    }
    cout<<"Enter the message to encrypt\n";
    string s;
    cin>>s;
    int temp = (n-s.size()%n)%n;
    for(i=0;i<temp;i++){
        s+='x';
    }
    k=0;
    string ans="";
    while(k<s.size()){
        for(i=0;i<n;i++){
            int sum = 0;
            int temp = k;
            for(j=0;j<n;j++){
                sum += (a[i][j]%26*(s[temp++]-'a')%26)%26;
                sum = sum%26;
            }
            ans+=(sum+'a');
        }
        k+=n;
    }
    cout<<ans<<"\n";

    return 0;
}
```

```

Hill cipher > C++ de.cpp > ...
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int modInverse(int a, int m)
6  {
7      a = a % m;
8      for (int x = -m; x < m; x++)
9          if ((a * x) % m == 1)
10             return x;
11 }
12
13 void getCofactor(vector<vector<int>> &a, vector<vector<int>> &temp, int p, int q, int n)
14 {
15     int i = 0, j = 0;
16     for (int row = 0; row < n; row++)
17     {
18         for (int col = 0; col < n; col++)
19         {
20             if (row != p && col != q)
21             {
22                 temp[i][j++] = a[row][col];
23                 if (j == n - 1)
24                 {
25                     j = 0;
26                 }
27             }
28         }
29     }
30 }
31
32 int main()
33 {
34     int n;
35     cout << "Enter the size of key matrix\n";
36     cin >> n;
37     vector<vector<int>> a(n, vector<int>(n));
38     cout << "Enter the key matrix\n";
39     for (int i = 0; i < n; i++)
40     {
41         for (int j = 0; j < n; j++)
42         {
43             cin >> a[i][j];
44         }
45     }
46     int det = 1;
47     for (int i = 0; i < n; i++)
48     {
49         det = det * a[i][i];
50     }
51     if (det % n != 0)
52     {
53         cout << "Inverse exist\n";
54     }
55     else
56     {
57         cout << "Inverse does not exist\n";
58     }
59     string message;
60     cout << "Enter the message to decrypt\n";
61     getline(cin, message);
62     string plainText = "";
63     for (int i = 0; i < message.length(); i++)
64     {
65         int x = message[i] - 'a';
66         int y = 0;
67         for (int j = 0; j < n; j++)
68         {
69             y = (x + a[i][j]) % n;
70         }
71         plainText += y;
72     }
73     cout << "Decrypted message is : " << plainText << endl;
74     return 0;
75 }

```

PROBLEMS 3 OUTPUT TERMINAL DEBUG CONSOLE

```

Enter the size of key matrix
2
Enter the key matrix
4 1
3 2
5 21
Inverse exist
Enter the message to decrypt
qzte
ram
PS D:\INS\Hill cipher>

```

## EX:

Plain Text : ram

Key : gybnqkcri

Ans : qzte

## 2. Decrypt following using Hill Cipher

### Code:

```

#include<iostream>
#include<vector>
using namespace std;

```

```

int modInverse(int a, int m){
    a=a%m;
    for(int x=-m;x<m;x++)
        if((a*x)%m==1)
            return x;
}

```

```

void getCofactor(vector<vector<int>> &a, vector<vector<int>> &temp, int p, int q, int n){
    int i=0,j=0;

```



```

for(int row=0;row<n;row++){
    for(int col=0;col<n;col++){
        if(row!=p&&col!=q){
            temp[i][j++] = a[row][col];
            if (j==n-1){
                j=0;
                i++;
            }
        }
    }
}
}
}

```

```

int determinant(vector<vector<int> > &a, int n, int N){
    int D = 0;
    if(n==1)
        return a[0][0];
    vector<vector<int> > temp(N, vector<int>(N));
    int sign = 1;
    for(int f=0;f<n;f++){
        getCofactor(a, temp, 0, f, n);
        D += sign * a[0][f] * determinant(temp, n - 1, N);
        sign = -sign;
    }
    return D;
}

```

```

void adjoint(vector<vector<int> > &a,vector<vector<int> > &adj,int N){
    if(N == 1){
        adj[0][0] = 1;
        return;
    }
    int sign = 1;
    vector<vector<int> > temp(N, vector<int>(N));
    for(int i=0;i<N;i++){
        for(int j=0;j<N;j++){
            getCofactor(a, temp, i, j, N);
            sign = ((i+j)%2==0)? 1: -1;
            adj[j][i] = (sign)*(determinant(temp, N-1 , N));
        }
    }
}

```

```

bool inverse(vector<vector<int> > &a, vector<vector<int> > &inv, int N){
    int det = determinant(a, N, N);
    if(det == 0){
        cout << "Inverse does not exist";
        return false;
    }
    int invDet = modInverse(det,26);
    cout<<det%26<<' '<<invDet<<'\n';
}

```

```

vector<vector<int> > adj(N, vector<int>(N));
adjoint(a, adj, N);
for(int i=0;i<N;i++)
    for(int j=0;j<N;j++)
        inv[i][j] = (adj[i][j]*invDet)%26;
return true;
}

int main() {
    int x,y,i,j,k,n;
    cout<<"Enter the size of key matrix\n";
    cin>>n;
    cout<<"Enter the key matrix\n";
    vector<vector<int> > a(n, vector<int>(n));
    vector<vector<int> > adj(n, vector<int>(n));
    vector<vector<int> > inv(n, vector<int>(n));

    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            cin>>a[i][j];
        }
    }
    if(inverse(a,inv,n)){
        cout<<"Inverse exist\n";
    }

    cout<<"Enter the message to decrypt\n";
    string s;
    cin>>s;
    k=0;
    string ans;
    while(k<s.size()){
        for(i=0;i<n;i++){
            int sum = 0;
            int temp = k;
            for(j=0;j<n;j++){
                sum += ((inv[i][j] + 26)%26*(s[temp++]-'a')%26)%26;
                sum = sum%26;
            }
            ans+=(sum+'a');
        }
        k+=n;
    }
    //ans+="\0";
    int f=ans.size()-1;
    while(ans[f]=='x'){
        f--;
    }

    for(i=0;i<=f;i++){
        cout<<ans[i];

```

```

    }
    cout<<"\n";
    return 0;
}

```

The screenshot shows a C++ IDE with two tabs: `code.cpp` and `de.cpp`. The `code.cpp` tab is active, displaying the following code:

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4  int main()
5  {
6      int x, y, i, j, k, n;
7      cout << "Enter the size of key matrix\n";
8      cin >> n;
9      cout << "Enter the key matrix\n";
10     int a[n][n];
11     for (i = 0; i < n; i++)
12     {
13         for (j = 0; j < n; j++)
14         {
15             cin >> a[i][j];
16         }
17     }
18     cout << "Enter the message to encrypt\n";
19     string s;
20     cin >> s;
21     int temp = (n - s.size() % n) % n;
22     for (i = 0; i < temp; i++)
23     {
24         s += 'x';
25     }

```

The terminal window shows the execution of the program:

```

63 | }
   | ^
PS D:\INS\Hill cipher> cd "d:\INS\Hill cipher\" ; if ($?) { g++ code.cpp -o code } ; if ($?) { .\code }
Enter the size of key matrix
2
Enter the key matrix
4 1
3 2
Enter the message to encrypt
ram
qzte
PS D:\INS\Hill cipher> cd "d:\INS\Hill cipher\" ; if ($?) { g++ de.cpp -o de } ; if ($?) { .\de }

```

**EX:**

Cipher Text : sok

Key : cabgxrthn

Ans : qzte