Output ⇒

Enter plain text :- hello

Enter key :- 3

khoor

Date:

Expt. No.: 1 (a)

Page No.:

Aim ⇒ WAP to implement Caesar Cipher cryptography.

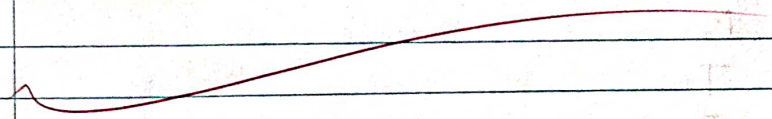
```
#include <stdio.h>
#include <conio.h>
#include <string.h>

void main() {
    char Str[10];
    int key, i;
    printf("Enter plain text :- ");
    gets(str);

    printf("\n Enter key :- ");
    scanf("%d", &key);

    while (key > 26 || key < -26) {
        key = key % 26;
    }

    for (i=0; i < strlen(str); i++) {
        if (str[i] <= 97 && str[i] >= 122)
            continue;
        str[i] += key;
        if (str[i] > 122) {
            int diff = str[i] - 122;
            str[i] = 'a' + diff - 1;
        }
```

Teacher's Signature:

Date: __/__/____

Expt. No.: ____ Page No.: ____

```
        if (str[i] < 97) {
            int diff = 97 - str[i];
            str[i] = 'z' - diff + 1;
        }
    }
    puts(str);
}
```



Teacher's Signature: _____

Output ⇒

Enter the key for playfair matrix :- Playfair

Playfair matrix :-

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

Enter the message for encryption :- Gaurav Verma
Modified string :- GAURAVVERMAX
Encoded Output :- HLVJWLUGGDWY
Decoded Output :- GAURAVVERMA

Date:

Expt. No.:- 1(b) Page No.:

Aim :- WAP to implement "Play Fair" Cryptography

```
l = []

# Generation of play fair matrix

def playfairmatrix():
    global l
    ip = (input("Enter the key for playfair
          matrix :- ").upper().replace
          ('J','I') + 'ABCDEFGHIKLMNO
          PQRSTUVWXYZ'
    s = ''
    for i in ip:
        if (i not in s) and (ord(i)>=65 and
            ord(i)<=90):
                s+=i

    p=0
    while (p<=20):
        m = []
        for i in range(0,5):
            m.append(s[i+p])
        l.append(m)
        p+=5
    print("Playfair Matrix :-")
    for i in l:
        print(i)
```

Teacher's Signature:

Date:

Expt. No.:

Page No.:

```
def checkval(a):
    x = -1
    for i in l:
        x += 1
        y = -1
        for j in i:
            y += 1
            if j == a:
                return x, y

def incval(v, y):
    if (v < 4)
        v += 1
    else:
        v = 0
    if (y < 4):
        y += 1
    else:
        y = 0
    return v, y

# Plain text to Cypher text
def encryption():
    ip = (input("Enter the message for encryption
        :- ").upper().replace('J','I')
```

Teacher's Signature:

Date: ___/___/____

Expt. No.: ____ Page No.: ____

```
op= ""
s = []
for i in ip:
    if (ord(i) >= 65 and ord(i) <= 90):
        s.append(i)
ec = ""
for i in s:
    ec += i
print("Modified string :- ", ec)
i = 0
while (i < len(s)):
    if (i == len(s)-1):
        s.append('z').
    if (s[i] == s[i+1]):
        s.insert(i+1, 'x')
    u, v = checkval(s[i])
    x, y = checkval(s[i+1])

    # other code than the repetation of char
    if (x == u):
        v, y = incrval(v, y)
        op += l[u][v]
        op += l[x][y]
    elif (v == y):
        u, x = incrval(u, x)
        op += l[u][v]
        op += l[x][y]
```

Teacher's Signature: _____

Date: __.__.____

Expt. No.: ____ Page No.: ____

```
        else :
            x, u = u, x
            op += l[x][y]
            op += l[u][v]
        i += 2
    print ("Encoded output :- ", op)
    return op

def decrval(v, y):
    if (v > 0):
        v -= 1
    else:
        v = 4
    if (y > 0):
        y -= 1
    else:
        y = 4
    return v, y

def decryption(ip):
    i = 0
    op = []
    while (i < len(ip)):
        u, v = checkval(ip[i])
        x, y = checkval(ip[i+1])
        if (x == 4):
```

Teacher's Signature: _____

Date: __.__.____

Expt. No.: ____ Page No.: ____

```
v,y = decrval(v,y)
op.append(l[u][v])
op.append(l[x][y])
elif(v==y):
    u,x = decrval(u,x)
    op.append(l[u][v])
    op.append(l[x][y])
else:
    x,u = u,x
    op.append(l[x][y])
    op.append(l[u][v])
if(len(op)>=4):
    if op[-2]==op[-4] and op[-3]=='x':
        op.remove('x')
    i+=2
if op[-1]=='z':
    op.pop()
msg=""
for i in op:
    msg+=i
print("Decoded string:-",msg)

playfairmatrix()
op=encryption()
decryption(op)
```

Teacher's Signature: _____

Output ⇒

```
Enter the size of key matrix
2
Enter the key matrix
4  1
3  2
Enter the message to encrypt:-
helloworld
gdddaivyvn
```

Date: __/__/____

Expt. No.: 1(c) Page No.: _____

Aim ⇒ WAP to implement Hill Cipher algorithm program in CPP

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

int main() {
    int n, 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 (int i=0; i<n; i++) {
        for (j=0; j<n; j++)
            cin >> a[i][j];
        cout << endl;
    }
    cout << "Enter the message to encrypt :- "
    string s;
    cin >> s;
    int temp = (n - s.size() % n) % n;
    for (i = 0; i < temp; i++) {
        s+ = 'x';
    }
    k = 0;
```

Teacher's Signature: _____

Expt. No.: Date:

Page No.:

```
string ans = " ";

while (k < s.size()) {
    for (i=0; i<n; i++) {
        int sum = 0;
        int temp = k;
        for (j=0; j<n; j++) {
            int sum = 0;
            int temp = k;
            for (j=0; j<n; j++) {
                sum += (a[i][j] % 26 * (s[temp+i]
                    - 'a') % 26) % 26;
                Sum = sum % 26;
            }
            ans += (sum + 'a');
        }
        k += n;
    }
    cout << ans << endl;
    return 0;
}
```

Teacher's Signature:

Date: ___/___/___

Expt. No.: 2 (a) Page No.: ____

Aim ⇒ WAP to implement Rail fence encryption and decryption program.

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

string encryptRailFence(string text, int key) {

    char rail[key][text.length()]
    for(int j = 0; j < text.length(); j++)
            rail[i][j] = '\n';
    bool dir_down = false;
    int row = 0, col = 0;
    for(int i = 0; i < text.length(); i++) {
        if(row == 0 || row == key - 1)
            dir_down = !dir_down;
        rail[row][col++] = text[i];
        dir_down ? row++ : row--;
    }
    string result;
    for(int i = 0; i < key; i++)
        for(int j = 0; j < text.length(); j++)
            if(rail[i][j] != '\n')
                result.push_back(rail[i][j]);
    return result;
}
```

Teacher's Signature: _____

Date: __/__/____

Expt. No.: ____ Page No.: ____

```
String decrypt Rail fence (String cipher, int key) {

    char rail [key] [cipher. length ()];

    for (int i=0; i< key ; i++)
        for (int j=0; j< cipher. length(); j++)
            rail [i] [j] = '\n';
    bool dir_down;
    int row = 0, col = 0;
    for (int i = 0 ; i< cipher. length (); i++)
    {
        if (row = = 0)
            dir_ down = true;
        if (row = = key - 1)
            dir_down = false;
        rail [row] [col++] = '*';
        dir_down ? row++ : row - -;
    }
    int index = 0;
    for (int i= 0; i< key; i++)
        for (int j=0; j< cipher. length (); j++)
            if (rail [i] [j] = = '*' &&
                    cipher. length ())
                rail [i] [j] = cipher [index++];
    string result;
    row = 0, col = 0;
```

Teacher's Signature: _____

Output ⇒

Enter message to encrypt :- attack at once
Encrypted msg = atc toctaka ne
Decrypted msg = attack at once

Date: __.__.____

Expt. No.: ____ Page No.: ____

```
for (int i=0; i< cipher.length(); i++)
{
    if (row == 0)
        dir_down = true;
    if (row == key-1)
        dir_down = false;
    if (rail[row][col] != '*')
        result.push_back(rail[row][col++]);
    dir_down ? row++ : row--;
}
return result;
}

int main()
{
    cout << "Enter message to encrypt :- " << endl;
    string msg;
    cin >> msg;
    string resultt = encryptRailFence(msg,3);
    cout << "Encrypted msg :- " << result << endl;
    cout << "Decrypted msg :- " << decryptRailFence
                                    (result,3);

    return 0;
}
```

Teacher's Signature: _____

Output :-

```
Enter the key:- crash
Enter the message :- Hello world
Encrypted message = lrhwodeoll
```

Date:

Expt. No.: 2 (b) Page No.:

Aim ⇒ WAP to implement the Row and column transposition cipher technique.

```
import math

key = input ("Enter the key:")

def encryption (msg):

    cipher = " "
    k_index = 0
    msg_len = len(msg)
    msg_list = list(msg)
    key_list = sorted (list(key))
    col = len(key)
    row = int (math.ceil (msg_len/col))
    fill_null = int ((row*col)- msg_len)
    msg_list.extended ('_' * fillnull)

    matrix = [msg_list [i : i+col]
                for i in range (0, len(msg_list, col)]

    for _ in range (col):
        curr_index = key.index (key_list [k_index])
        cipher += ''.join ([row [curr_idn]
                            for row in matrix])
        k_index += 1
```

Teacher's Signature:

Date: _____

Expt. No.: _____ Page No.: _____

```
ceasar cipher

msg = input ("Enter the message")
print ("Encrypted message :-" + encryption(msg))
```

Teacher's Signature: _____

Date: __-__-____

Expt. No.: 3. Page No.: ____

Aim :- WAP to implement Diffie Hellman Key Exchange Algorithm

```
#include <stdio.h>
#include <math.h>

long long int power (long long int a, long long int b,
                     long long int P)
{
    if (b == 1)
        return a;
    else
        return (((long long int) pow(a, b)) % P);
}
Void main () {

    long long int P, G, x, a, y, b, ka, kb;
    P = 23;
    print ("The value of P: %lld \n", P);
    G = 9;
    printf ("The value of G: %lld \n", G);
    a = 4
    printf ("The value of key a : %lld\n", a);
    x = power (G, a, P);
    b = 3;
    print f (" The value of key b: %lld \n", b);
```

Teacher's Signature: _____

Output ⇒

The value of P: 23

The value of G: 9

The private key a : 4

The private key b: 3

secret key for the Alice is : 9

secret key for the Bob is : 9

Date: _____

Expt. No.: _____ Page No.: _____

```
y = power (G, b, P);

ka = power (y, a, P);
kb = power (x, b, P);

printf ("Secret key for the Alice is : %.lld \n", ka);
printf ("Secret key for the Bob is : %.lld \n", kb);
```

2 | 3 | 10 | 22

Teacher's Signature: _____