Experiment no:- 01

```python
# Function to encrypt the text using Caesar Cipher
def caesar_encrypt(plain_text, shift):
    encrypted_text = ""
    for char in plain_text:
        if char.isalpha():
            shift_amount = shift % 26
            char_code = ord(char) + shift_amount
            if char.islower():
                if char_code > ord('z'):
                    char_code -= 26
                encrypted_text += chr(char_code)
            elif char.isupper():
                if char_code > ord('Z'):
                    char_code -= 26
                encrypted_text += chr(char_code)
        else:
            encrypted_text += char
    return encrypted_text

# Function to decrypt the text using Caesar Cipher
def caesar_decrypt(cipher_text, shift):
    return caesar_encrypt(cipher_text, -shift)

# Main function to interact with the user
if __name__ == "__main__":  # Corrected this line
    # Get user input for plain text
    text = input("Enter the text: ")

    # Get user input for the shift key
    while True:
        try:
            shift_key = int(input("Enter the shift key (0-25): "))
```

```python
    while True:
        try:
            shift_key = int(input("Enter the shift key (0-25): "))
            if 0 <= shift_key <= 25:
                break
            else:
                print("Please enter a valid key between 0 and 25.")
        except ValueError:
            print("Please enter a valid integer.")

    # Encrypt the text
    encrypted = caesar_encrypt(text, shift_key)
    print(f"Encrypted Text: {encrypted}")

    # Ask user if they want to decrypt
    decrypt_choice = input("Do you want to decrypt the text? (y/n): ").strip().lower()
    if decrypt_choice == 'y':
        while True:
            try:
                decryption_key = int(input("Enter the decryption key: "))
                break
            except ValueError:
                print("Please enter a valid integer.")
    if decrypt_choice == 'y':
        decrypted = caesar_decrypt(encrypted, shift_key)
        print(f"Decrypted Text: {decrypted}")
    else:
        print("Decryption skipped.")
```

Output:

```
PS F:\Users\User\Desktop\Experiments> python -u "f:\Users\User\Desktop\Experiments\Jaykit\Caesar.py"
Enter the text: jaykit
Enter the shift key (0-25): 3
Encrypted Text: mdbnlw
Do you want to decrypt the text? (y/n): y
Enter the decryption key: 3
Decrypted Text: jaykit
PS F:\Users\User\Desktop\Experiments>
```

Experiment no:- 03

```python
def create_matrix(key):
    key = ''.join(sorted(set(key), key=key.index))
    alphabet = "abcdefghiklmnopqrstuvwxyz"
    matrix = []

    for char in key:
        if char not in matrix and char != 'j':
            matrix.append(char)

    for char in alphabet:
        if char not in matrix:
            matrix.append(char)

    return [matrix[i:i + 5] for i in range(0, 25, 5)]


def format_plaintext(plain_text):
    plain_text = plain_text.replace('j', 'i')
    formatted_text = ""
    i = 0

    while i < len(plain_text):
        char1 = plain_text[i]
        if i + 1 < len(plain_text):
            char2 = plain_text[i + 1]
            if char1 == char2:
                formatted_text += char1 + 'x'
                i += 1
            else:
                formatted_text += char1 + char2
                i += 2
        else:
```

```python
def format_plaintext(plain_text):
                formatted_text += char1 + char2
                i += 2
            else:
                formatted_text += char1 + 'x'
                i += 1

    return formatted_text


def find_position(matrix, char):
    for row in range(5):
        for col in range(5):
            if matrix[row][col] == char:
                return row, col
    return None


def encrypt_pair(matrix, char1, char2):
    row1, col1 = find_position(matrix, char1)
    row2, col2 = find_position(matrix, char2)

    if row1 == row2:
        return matrix[row1][(col1 + 1) % 5] + matrix[row2][(col2
    elif col1 == col2:
        return matrix[(row1 + 1) % 5][col1] + matrix[(row2 + 1) %
    else:
        return matrix[row1][col2] + matrix[row2][col1]


def decrypt_pair(matrix, char1, char2):
    row1, col1 = find_position(matrix, char1)
```

```python
def decrypt_pair(matrix, char1, char2):
    row1, col1 = find_position(matrix, char1)
    row2, col2 = find_position(matrix, char2)

    if row1 == row2:
        return matrix[row1][(col1 - 1) % 5] + matrix[row2][(col2
    elif col1 == col2:
        return matrix[(row1 - 1) % 5][col1] + matrix[(row2 - 1) %
    else:
        return matrix[row1][col2] + matrix[row2][col1]


def playfair_encrypt(plain_text, key):
    matrix = create_matrix(key)
    plain_text = format_plaintext(plain_text)
    encrypted_text = ""

    for i in range(0, len(plain_text), 2):
        encrypted_text += encrypt_pair(matrix, plain_text[i], pla

    return encrypted_text


def playfair_decrypt(cipher_text, key):
    matrix = create_matrix(key)
    decrypted_text = ""

    for i in range(0, len(cipher_text), 2):
        decrypted_text += decrypt_pair(matrix, cipher_text[i], ci

    return decrypted_text
```

```python

def validate_key(key):
    if key.isalpha():
        return True
    else:
        print("Invalid Key: The key must contain only alphabetic
        return False

########################### Main function#######################
def main():
    while True:
        choice = input("Enter 'e' to Encrypt or 'd' to Decrypt (o
        if choice == 'q':
            break

        text = input("Enter the text: ").lower().replace(" ", "")
        key = input("Enter the key (Alphabetic Only): ").lower().

        if validate_key(key):
            if choice == 'e':
                encrypted_text = playfair_encrypt(text, key)
                print(f"Cipher Text: {encrypted_text}\n")
            elif choice == 'd':
                decrypted_text = playfair_decrypt(text, key)
                print(f"Decrypted Text: {decrypted_text}\n")
            else:
                print("Invalid choice. Please enter 'e' to Encrypt

if __name__ == "__main__":
    main()
```

Output:

```
PS F:\Users\User\Desktop\Experiments> python -u "f:\Users\User\Desktop\Experiments\Jaykit\playfair.py"
Enter 'e' to Encrypt or 'd' to Decrypt (or 'q' to quit): e
Enter the text: MAURYA
Enter the key (Alphabetic Only): APPLE
Cipher Text: ilqsve

Enter 'e' to Encrypt or 'd' to Decrypt (or 'q' to quit): d
Enter the text: ilqsve
Enter the key (Alphabetic Only): APPLE
Decrypted Text: maurya

Enter 'e' to Encrypt or 'd' to Decrypt (or 'q' to quit):
```

Experiment no:- 02

```python

def caesar_encrypt(plain_text, shift):
    encrypted_text = ""
    for char in plain_text:
        if char.isalpha():
            shift_amount = shift % 26
            char_code = ord(char) + shift_amount
            if char.islower():
                if char_code > ord('z'):
                    char_code -= 26
                encrypted_text += chr(char_code)
            elif char.isupper():
                if char_code > ord('Z'):
                    char_code -= 26
                encrypted_text += chr(char_code)
        else:
            encrypted_text += char
    return encrypted_text

# Function to decrypt the text using Caesar Cipher with a given s
def caesar_decrypt(cipher_text, shift):
    decrypted_text = ""
    for char in cipher_text:
        if char.isalpha():
            shift_amount = shift % 26
            char_code = ord(char) - shift_amount
            if char.islower():
                if char_code < ord('a'):
                    char_code += 26
                decrypted_text += chr(char_code)
            elif char.isupper():
                if char_code < ord('A'):
```

```python
def caesar_decrypt(cipher_text, shift):
                if char_code < ord('A'):
                    char_code += 26
                decrypted_text += chr(char_code)
            else:
                decrypted_text += char
    return decrypted_text

# Function to perform brute-force cryptanalysis on Caesar Cipher
def brute_force_caesar(cipher_text):
    possible_plaintexts = []
    for shift in range(1, 26):
        decrypted_text = caesar_decrypt(cipher_text, shift)
        possible_plaintexts.append((shift, decrypted_text))
    # Print out each possible decryption
    for shift, text in possible_plaintexts:
        print(f"Shift {shift}: {text}")

    return possible_plaintexts

def main():

    cipher_text = "Khoor Zruog!"

    print("Attempting brute-force decryption:")
    brute_force_caesar(cipher_text)

if __name__ == "__main__":
    main()
```

Output:

```python
def brute_force_caesar(cipher_text):
        print(f"Shift {shift}: {text}")

    return possible_plaintexts
```

```
PS F:\Users\User\Desktop\Experiments> python -u "f:\Users\User\Desktop\Experiments\Jaykit\main.py"
Attempting brute-force decryption:
Shift 1: Jgnnq Yqtnf!
Shift 2: Ifmmp Xpsme!
Shift 3: Hello World!
Shift 4: Gdkkn Vnqkc!
Shift 5: Fcjjm Umpjb!
Shift 6: Ebiil Tloia!
Shift 7: Dahhk Sknhz!
Shift 8: Czggj Rjmgy!
Shift 9: Byffi Qilfx!
Shift 10: Axeeh Phkew!
Shift 11: Zwddg Ogjdv!
Shift 12: Yvccf Nficu!
Shift 13: Xubbe Mehbt!
Shift 14: Wtaad Ldgas!
Shift 15: Vszzc Kcfzr!
Shift 16: Uryyb Jbeyq!
Shift 17: Tqxxa Iadxp!
Shift 18: Spwwz Hzcwo!
Shift 19: Rovvy Gybvn!
Shift 20: Qnuux Fxaum!
Shift 21: Pmttw Ewztl!
Shift 22: Olssv Dvysk!
Shift 23: Nkrru Cuxrj!
Shift 24: Mjqqt Btwqi!
Shift 25: Lipps Asvph!
PS F:\Users\User\Desktop\Experiments>
```

Experiment no:- 04

1)ipconfig

```
C:\WINDOWS\system32\cmd.

Microsoft Windows [Version 10.0.22631.4112]
(c) Microsoft Corporation. All rights reserved.

C:\Users\JAYKIT>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2402:3a80:4179:b3cd:84bc:5750:a750:51f7
   Temporary IPv6 Address. . . . . . : 2402:3a80:4179:b3cd:7107:83d2:85f3:159c
   Link-local IPv6 Address . . . . . : fe80::977b:dc69:568f:49e%5
   IPv4 Address. . . . . . . . . . . : 192.168.10.60
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::40e7:9dff:fe74:e944%5

C:\Users\JAYKIT>
```

2)ipconfig /all

```
C:\WINDOWS\system32\cmd.

C:\Users\JAYKIT>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : DESKTOP-RB7SF1Q
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physical Address. . . . . . . . . : 6C-02-E0-CD-BC-40
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection* 1:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter
   Physical Address. . . . . . . . . : DA-C0-A6-F5-7D-15
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Microsoft Wi-Fi Direct Virtual Adapter #2
   Physical Address. . . . . . . . . : D8-C0-A6-F5-7D-15
   DHCP Enabled. . . . . . . . . . . : No
```

3)netstat

```
C:\WINDOWS\system32\cmd.

C:\Users\JAYKIT>netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    127.0.0.1:49671        DESKTOP-RB7SF1Q:49672  ESTABLISHED
  TCP    127.0.0.1:49672        DESKTOP-RB7SF1Q:49671  ESTABLISHED
  TCP    127.0.0.1:49673        DESKTOP-RB7SF1Q:49674  ESTABLISHED
  TCP    127.0.0.1:49674        DESKTOP-RB7SF1Q:49673  ESTABLISHED
  TCP    127.0.0.1:49862        DESKTOP-RB7SF1Q:49863  ESTABLISHED
  TCP    127.0.0.1:49863        DESKTOP-RB7SF1Q:49862  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50280  [2603:1040:a06:6::1]:https  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50394  [64:ff9b::14a7:52e1]:https  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50398  [2603:1040:a06:6::]:https  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50403  sh-in-f188:5228       ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50408  whatsapp-chatd-edge6-shv-01-atl3:http  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50409  [2a04:4e42:4e::684]:http  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50410  [2a04:4e42:4f::684]:http  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50518  [2620:1ec:46::254]:https  CLOSE_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50520  bom12s06-in-x0e:https  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50532  bom07s33-in-x0e:https  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50566  [64:ff9b::14d4:5875]:https  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50568  g2600-1417-0077-0000-0000-0000-170b-d637:https  CLOSE_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50570  [2603:1046:c04:83b::2]:https  ESTABLISHED
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50573  g2600-1417-0077-0000-0000-0000-170b-d649:https  CLOSE_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50600  maa03s46-in-x01:https  TIME_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50601  maa03s41-in-x02:https  TIME_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50602  [64:ff9b::8efa:c342]:https  TIME_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50604  maa05s12-in-x01:https  TIME_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50605  bom05s10-in-x01:https  TIME_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50606  maa03s31-in-f6:https  TIME_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50612  bom07s10-in-x03:https  CLOSE_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50618  [64:ff9b::3468:8335]:https  TIME_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50620  [64:ff9b::3384:c169]:https  TIME_WAIT
  TCP    [2402:3a80:4179:b3cd:7107:83d2:85f3:159c]:50623  [64:ff9b::8efa:43e3]:https  ESTABLISHED
```

4)ping

```
C:\WINDOWS\system32\cmd.

C:\Users\JAYKIT>ping

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]
            [-4] [-6] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet (IPv4-only).
    -i TTL         Time To Live.
    -v TOS         Type Of Service (IPv4-only. This setting has been deprecated
                   and has no effect on the type of service field in the IP
                   Header).
    -r count       Record route for count hops (IPv4-only).
    -s count       Timestamp for count hops (IPv4-only).
    -j host-list   Loose source route along host-list (IPv4-only).
    -k host-list   Strict source route along host-list (IPv4-only).
    -w timeout     Timeout in milliseconds to wait for each reply.
    -R             Use routing header to test reverse route also (IPv6-only).
                   Per RFC 5095 the use of this routing header has been
                   deprecated. Some systems may drop echo requests if
                   this header is used.
    -S srcaddr     Source address to use.
    -c compartment Routing compartment identifier.
    -p             Ping a Hyper-V Network Virtualization provider address.
    -4             Force using IPv4.
    -6             Force using IPv6.
```

5)tracert

```
C:\Users\JAYKIT>tracert

Usage: tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout]
               [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -d                 Do not resolve addresses to hostnames.
    -h maximum_hops    Maximum number of hops to search for target.
    -j host-list       Loose source route along host-list (IPv4-only).
    -w timeout         Wait timeout milliseconds for each reply.
    -R                 Trace round-trip path (IPv6-only).
    -S srcaddr         Source address to use (IPv6-only).
    -4                 Force using IPv4.
    -6                 Force using IPv6.
```

6)ping

```
C:\Users\JAYKIT>ping google.com

Pinging google.com [2404:6800:4009:820::200e] with 32 bytes of data:
Reply from 2404:6800:4009:820::200e: time=31ms
Reply from 2404:6800:4009:820::200e: time=34ms
Reply from 2404:6800:4009:820::200e: time=34ms
Reply from 2404:6800:4009:820::200e: time=34ms

Ping statistics for 2404:6800:4009:820::200e:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 31ms, Maximum = 34ms, Average = 33ms
```

7)arp -a

```
C:\Users\JAYKIT>arp -a

Interface: 192.168.10.60 --- 0x5
  Internet Address      Physical Address      Type
  192.168.10.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

8)nslookup

```
C:\Users\JAYKIT>nslookup
DNS request timed out.
    timeout was 2 seconds.
Default Server:  UnKnown
Address:  2402:3a80:4179:b3cd::81

> www.google.com
Server:  UnKnown
Address:  2402:3a80:4179:b3cd::81

Non-authoritative answer:
Name:    www.google.com
Addresses:  2404:6800:4007:823::2004
          142.250.182.196
```

8)tracert 8.8.8.8

```
C:\Users\JAYKIT>tracert 8.8.8.8

Tracing route to dns.google [8.8.8.8]
over a maximum of 30 hops:

  1  Transmit error: code 1231.

Trace complete.
```

9) route

```
Command Prompt

C:\Users\JAYKIT>route

Manipulates network routing tables.

ROUTE [-f] [-p] [-4|-6] command [destination]
                  [MASK netmask]  [gateway] [METRIC metric]  [IF interface]

  -f           Clears the routing tables of all gateway entries.  If this is
               used in conjunction with one of the commands, the tables are
               cleared prior to running the command.

  -p           When used with the ADD command, makes a route persistent across
               boots of the system. By default, routes are not preserved
               when the system is restarted. Ignored for all other commands,
               which always affect the appropriate persistent routes.

  -4           Force using IPv4.

  -6           Force using IPv6.

  command      One of these:
                 PRINT     Prints  a route
                 ADD       Adds    a route
                 DELETE    Deletes a route
                 CHANGE    Modifies an existing route
  destination  Specifies the host.
  MASK         Specifies that the next parameter is the 'netmask' value.
  netmask      Specifies a subnet mask value for this route entry.
               If not specified, it defaults to 255.255.255.255.
  gateway      Specifies gateway.
  interface    the interface number for the specified route.
  METRIC       specifies the metric, ie. cost for the destination.

All symbolic names used for destination are looked up in the network database
```

```
Command Prompt

If the command is PRINT or DELETE. Destination or gateway can be a wildcard,
(wildcard is specified as a star '*'), or the gateway argument may be omitted.

If Dest contains a * or ?, it is treated as a shell pattern, and only
matching destination routes are printed. The '*' matches any string,
and '?' matches any one char. Examples: 157.*.1, 157.*, 127.*, *224*.

Pattern match is only allowed in PRINT command.
Diagnostic Notes:
    Invalid MASK generates an error, that is when (DEST & MASK) != DEST.
    Example> route ADD 157.0.0.0 MASK 155.0.0.0 157.55.80.1 IF 1
             The route addition failed: The specified mask parameter is invalid. (Destination & Mask) != Destination.

Examples:

    > route PRINT
    > route PRINT -4
    > route PRINT -6
    > route PRINT 157*          .... Only prints those matching 157*

    > route ADD 157.0.0.0 MASK 255.0.0.0  157.55.80.1 METRIC 3 IF 2
            destination^      ^mask      ^gateway     metric^    ^
                                                        Interface^
      If IF is not given, it tries to find the best interface for a given
      gateway.
    > route ADD 3ffe::/32 3ffe::1

    > route CHANGE 157.0.0.0 MASK 255.0.0.0 157.55.80.5 METRIC 2 IF 2

      CHANGE is used to modify gateway and/or metric only.

    > route DELETE 157.0.0.0
    > route DELETE 3ffe::/32
```