Write a Python program to implement the Caesar cipher using Substitution technique.

```
def encrypt(text, s):
    result = ""
    for char in text:
        if char.isupper():
            result += chr((ord(char) + s - 65) % 26 + 65)
        elif char.islower():

            result += chr((ord(char) + s - 97) % 26 + 97)
        else:
            result += char
    return result

text = "FAMT"
s = 4
print("Text : " + text)
print("Shift : " + str(s))
print("Cipher: " + encrypt(text, s))
```

Write a Python program to implement and analysis of RSA.

```
from sympy import isprime, gcd, mod_inverse

# Step 1: Key Generation
def generate_keys(p, q, e):
    if not (isprime(p) and isprime(q)):
        raise ValueError("Both numbers must be prime.")
    if p == q:
        raise ValueError("p and q cannot be the same.")

    n = p * q
    phi_n = (p - 1) * (q - 1)

    if gcd(e, phi_n) != 1:
        raise ValueError(f"e = {e} is not coprime with φ(n) = {phi_n}.
Choose a different e.")

    d = mod_inverse(e, phi_n)

    print(f"\nComputed values:")
```

```
    print(f"n = {n}")
    print(f"φ(n) = {phi_n}")
    print(f"Private key d = {d}")

    return (e, d, n)

# Step 2: Encryption
def encrypt(message, e, n):
    return pow(message, e, n)

# Step 3: Decryption
def decrypt(cipher, d, n):
    return pow(cipher, d, n)

# Main Program
if __name__ == "__main__":
    try:
        p = int(input("Enter a prime number p: "))
        q = int(input("Enter a different prime number q: "))
        e = int(input("Enter public exponent e (coprime with φ(n)): "))

        # Key generation
        e, d, n = generate_keys(p, q, e)

        # Message input
        message = int(input("\nEnter a number message to encrypt (as
integer): "))

        # Encryption
        cipher = encrypt(message, e, n)
        print(f"Encrypted message: {cipher}")

        # Decryption
        decrypted = decrypt(cipher, d, n)
        print(f"Decrypted message: {decrypted}")

    except ValueError as ve:
        print(f"Error: {ve}")
```

Enter p=5,q=7,n=11,enc=2

Write a Python program to implement the Diffie-Hellman Key Exchange Algorithm.

```
n = int(input("Enter prime no (n): "))
g = int(input("Enter primitive root (g): "))
x = int(input("Enter private key of Alice (x): "))
y = int(input("Enter private key of Bob (y): "))
A = pow(g, x, n)
print(f"Alice's Public key (A) = {A}")
B = pow(g, y, n)
print(f"Bob's Public key (B) = {B}")
k1 = pow(B, x, n)
print(f"Alice's Computed Shared key (k1) = {k1}")
k2 = pow(A, y, n)
print(f"Bob's Computed Shared key (k2) = {k2}")
if k1 == k2:
    print("Key exchange Successful!")
    print(f"Shared secret key = {k1}")
else:
    print("Key exchange Failed!")
```

n=7,g=4,x=3,y=6,a=6,b=16,

Write a Python program to implement MD5 Algorithm

```
import hashlib

def generatemd5message(message):
    md5hash = hashlib.md5()
    md5hash.update(message.encode('utf-8'))
    return md5hash.hexdigest()

text = input("Enter text to hash using MD5: ")

md5result = generatemd5message(text)

print("Original text:", text)
print("MD5 Hash:", md5result)
```

mayuresh

Write a Python program to implement SHA Algorithm

```
import hashlib

def generateshalmessage(message):
    shalhash = hashlib.sha1()
    shalhash.update(message.encode('utf-8'))
    return shalhash.hexdigest()
text = input("Enter text to hash using SHA-1: ")

shalresult = generateshalmessage(text)

print("Original text:", text)
print("SHA-1 Hash:", shalresult)
```

Mayuresh

Implement a code to simulate DOS attack.

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

int main(void)
{
   char buff[15]; // buffer size = 15 bytes
   int pass = 0;
   printf("\n Enter the password: \n");
   gets(buff); // ?? UNSAFE: no bounds checking

   if (strcmp(buff, "thecorrectpaswd")) {
      printf("\n Wrong Password \n");
   }
   else {
      printf("\n Correct Password \n");
      pass = 1;
   }
   if (pass) {
      printf("\n Root privileges given to the user \n");
   }
   return 0;
}
```

output:C:\Users\dell\Documents\security lab>gcc p8.c -o p8

C:\Users\dell\Documents\security lab>p8.exe

Enter the password:
thecorrectpaswd

Correct Password

Root privileges given to the user

C:\Users\dell\Documents\security lab>p8.exe

Enter the password:
vbjfdbvk

Wrong Password

C:\Users\dell\Documents\security lab>