Name: Tushar Panchal

En.No: 21162101014

Sub: INS (INFORMATION SECURITY)

Branch: CBA

Batch:61

---------------------------PRACTICAL 10---------------------------

❖ AIM :

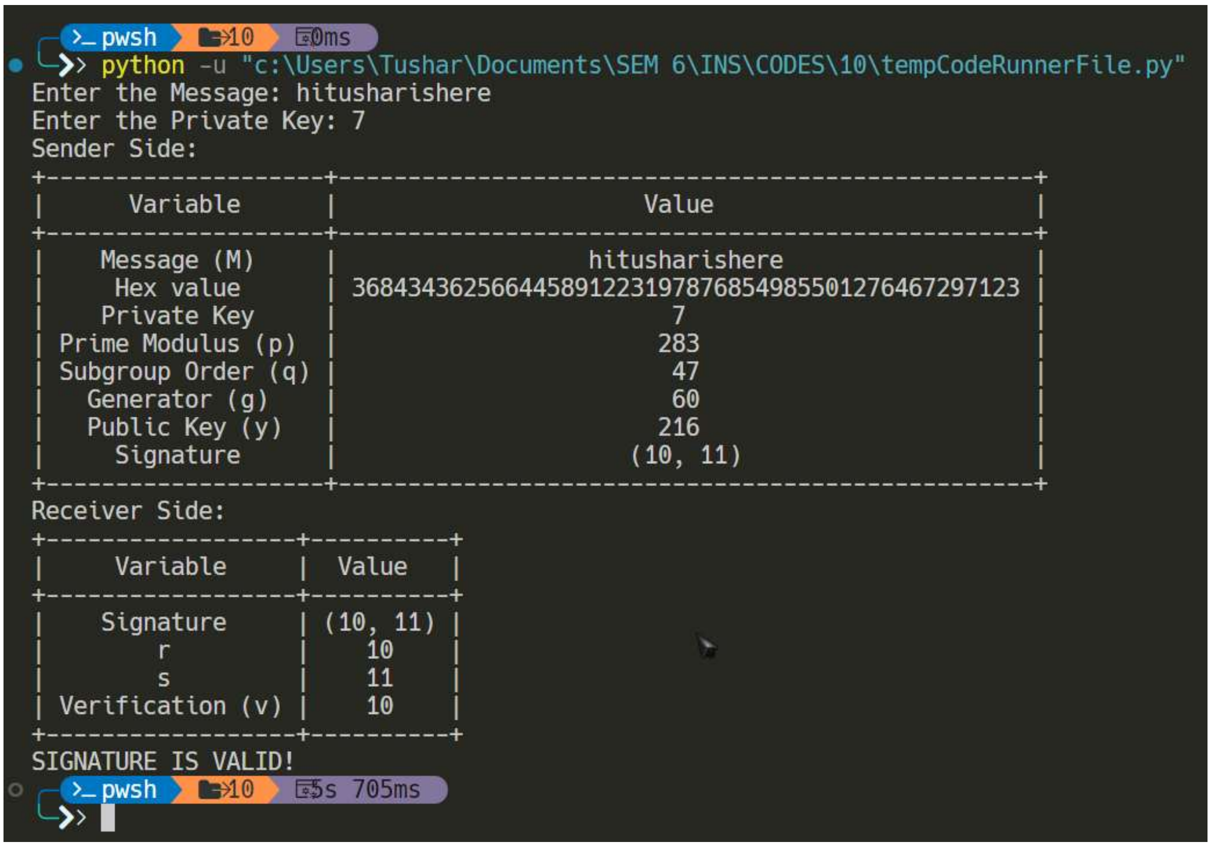
Consider a scenario where in a company two employee wants to authenticate them self as legitimate entity. Provide a solution for authentication of two parties through digital signature.

✓ Source Code :

```
from hashlib import sha1
from sympy import mod_inverse
from prettytable import PrettyTable

def generate_public_key(base, prime_mod, private_key):
    public_key = pow(base, private_key, prime_mod)
    return public_key

def generate_signature(base, prime_mod, subgroup_order, hashed_message,
secret_key, private_key):
    r = pow(base, secret_key, prime_mod) % subgroup_order
    secret_inverse = mod_inverse(secret_key, subgroup_order)
    s = (hashed_message + private_key * r) * secret_inverse %
subgroup_order
    return (r, s)

def verify_signature(signature_r, signature_s, prime_mod,
subgroup_order, hashed_message, base, public_key):
    s_inverse = mod_inverse(signature_s, subgroup_order)
    u1 = (hashed_message * s_inverse) % subgroup_order
    u2 = (signature_r * s_inverse) % subgroup_order
    v = (pow(base, u1, prime_mod) * pow(public_key, u2, prime_mod)) %
prime_mod % subgroup_order
    return v

def hex_digest(message):
    return int(sha1(message.encode()).hexdigest(), 16)

prime_modulus = 283
subgroup_order = 47
generator = 60

message = input("Enter the Message: ")
private_key = int(input("Enter the Private Key: "))

hashed_message = hex_digest(message)
xr = private_key % subgroup_order
public_key = generate_public_key(generator, prime_modulus, xr)

random_k = 43 % subgroup_order
signature = generate_signature(generator, prime_modulus,
subgroup_order, hashed_message, random_k, xr)
r, s = signature

verification = verify_signature(r, s, prime_modulus, subgroup_order,
hashed_message, generator, public_key)

sender_table = PrettyTable()
sender_table.field_names = ['Variable', 'Value']
sender_table.add_rows([
    ["Message (M)", message],
    ["Hex value", hashed_message],
    ["Private Key", private_key],
    ["Prime Modulus (p)", prime_modulus],
    ["Subgroup Order (q)", subgroup_order],
    ["Generator (g)", generator],
    ["Public Key (y)", public_key],
    ["Signature", signature]
])

receiver_table = PrettyTable()
receiver_table.field_names = ['Variable', 'Value']
receiver_table.add_rows([
    ['Signature', signature],
    ['r', r],
    ['s', s],
    ['Verification (v)', verification]
])
```

```python
print("Sender Side:")
print(sender_table)

print("Receiver Side:")
print(receiver_table)

if verification == r:
    print("SIGNATURE IS VALID!")
else:
    print("SIGNATURE IS INVALID!")
```

- ✓ **Output :**



```
 >_ pwsh  10  0ms
╰─>> python -u "c:\Users\Tushar\Documents\SEM 6\INS\CODES\10\tempCodeRunnerFile.py"
Enter the Message: hitusharishere
Enter the Private Key: 7
Sender Side:
+--------------------+--------------------------------------------------+
|      Variable      |                      Value                       |
+--------------------+--------------------------------------------------+
|    Message (M)     |                  hitusharishere                  |
|     Hex value      | 368434362566445891223197876854985501276467297123 |
|    Private Key     |                        7                         |
| Prime Modulus (p)  |                       283                        |
| Subgroup Order (q) |                        47                        |
|   Generator (g)    |                        60                        |
|   Public Key (y)   |                       216                        |
|     Signature      |                     (10, 11)                     |
+--------------------+--------------------------------------------------+
Receiver Side:
+------------------+----------+
|     Variable     |  Value   |
+------------------+----------+
|    Signature     | (10, 11) |
|        r         |    10    |
|        s         |    11    |
| Verification (v) |    10    |
+------------------+----------+
SIGNATURE IS VALID!
 >_ pwsh  10  5s 705ms
╰─>> 
```