Assignment
Cryptography
Name: Smraddhi Rathore
Roll No.: 2021bcy0025

```python
import random
def is_prime(n):
    if n <= 1:
        return False
    elif n <= 3:
        return True
    elif n % 2 == 0 or n % 3 == 0:
        return False
    i = 5
    while i * i <= n:
        if n % i == 0 or n % (i + 2) == 0:
            return False
        i += 6
    return True
def primitive_root(p):
    primitive_roots = []
    for a in range(2, p):
        if pow(a, p - 1, p) == 1:
            primitive_roots.append(a)
    return primitive_roots
while True:
    p = random.randint(2**10, 2**12)
    if is_prime(p):
        break
primitive_roots = primitive_root(p)
g = random.choice(primitive_roots)
print("Prime number (p):", p)
print("Primitive root (g):", g)
a = random.randint(2, p - 1)
A = pow(g, a, p)
b = random.randint(2, p - 1)
B = pow(g, b, p)
s_Alice = pow(B, a, p)
s_Bob = pow(A, b, p)
message = "Is this encoded›‹ ›»"
print("Original message:", message)
otp_key = random.randint(0, 2**len(message)-1)
print("One-time pad key:", otp_key)
```

```
cipher_text = ''
for i in range(len(message)):
    cipher_text += chr((ord(message[i]) + otp_key) % 256)
plain_text = ''
for i in range(len(cipher_text)):
    plain_text += chr((ord(cipher_text[i]) - otp_key) % 256)
print("Decrypted message:", plain_text)
```

```
smraddhi@smraddhis-MacBook-Air Cryptography % python -u "/Users/smraddhi/Documents/Cryptography/assignment.py"
Prime number (p): 1531
Primitive root (g): 762
Original message: Is this encoded
One-time pad key: 10931
Decrypted message: Is this encoded
smraddhi@smraddhis-MacBook-Air Cryptography %
```