11) To write a python program to implement the signature scheme named digital signature standard (Euclidean Algorithm).

PROGRAM:-

```
import random
one = 1
zero = 0
def isprime(n):
    if n <= 1:
        return False
    if n == 2 or n == 3:
        return True
    if 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 get_next_prime(ans):
    test = int(ans)
    while not isprime(test):
        test += 1
    return test
def find_q(n):
    start = 2
    while not isprime(n):
        while n % start != 0:
            start += 1
        n = n // start
    return n
```

```
def get_gen(p, q, rand_obj):
    h = random.randint(1, p-1)
    return pow(h, (p - 1) // q, p)
def main():
    try:
        rand_obj = random
        p = get_next_prime(10600)
        q = find_q(p - 1)
        g = get_gen(p, q, rand_obj)
        print(f"p: {p}\nq: {q}\ng: {g}")
        x = random.randint(1, q-1)
        y = pow(g, x, p)
        k = random.randint(1, q-1)
        r = pow(g, k, p) % q
        hash_val = random.randint(1, p-1)
        k_inv = pow(k, -1, q)
        s = (k_inv * (hash_val + x * r)) % q
        print(f"\nSignature (r, s):\n r: {r}\ns: {s}")
        w = pow(s, -1, q)
        u1 = (hash_val * w) % q
        u2 = (r * w) % q
        v = (pow(g, u1, p) * pow(y, u2, p)) % p % q
        print(f"\nVerification:\nw: {w}\nu1: {u1}\nu2: {u2}\nv: {v}")
        if v == r:
            print("\nSuccess: Signature verified!")
        else:
            print("\nError: Invalid signature.")
    except Exception as e:
        print("Error:", e)
if __name__ == "__main__":
    main()
```

OUTPUT:-

```
p: 10601
q: 53
g: 29

Signature (r, s):
 r: 36
s: 29

Verification:
w: 11
u1: 40
u2: 25
v: 36

Success: Signature verified!
```