32. Write a python program for DSA, because the value of k is generated for each signature, even if the same message is signed twice on different occasions, the signatures will differ. This is not true of RSA signatures. Write a C program for implication of this difference?

--

Simple demonstration that DSA signatures change each time

because k is chosen randomly for each signature.

(Toy math, not secure. For teaching only.)

--

Code:

```
import random

# toy small parameters

p = 30803

q = 101

g = 2

# private key

x = 45

y = pow(g, x, p)

def dsa_sign(message):

   # random per-signature k

   k = random.randint(1, q - 1)

    r = pow(g, k, p) % q

   k_inv = pow(k, -1, q)         # modular inverse

   s = (k_inv * (message + x * r)) % q

   return (r, s)

msg = 12345

print("Signing same message twice with DSA:\n")

sig1 = dsa_sign(msg)

sig2 = dsa_sign(msg)
```

```
print("Signature 1:", sig1)

print("Signature 2:", sig2)

if sig1 != sig2:

    print("\nAs expected, signatures differ because k is random.")
```

```
>>>
========== RESTART: C:/Users/Maria/OneDrive/Documents/ex32.py ==========
Signing same message twice with DSA:

Signature 1: (68, 88)
Signature 2: (98, 52)

As expected, signatures differ because k is random.
>>>
```