27. Write a python program for Bob uses the RSA cryptosystem with a very large modulus n for which the factorization cannot be found in a reasonable amount of time. Suppose Alice sends a message to Bob by representing each alphabetic character as an integer between 0 and 25 (A S 0, c, Z S 25) and then encrypting each number separately using RSA with large e and large n. Is this method secure? If not, describe the most efficient attack against this encryption method.

Code:

```
import math

# Bob’s original RSA keys (public = e, private = d)

n = 3599       # Bob's modulus

e = 31         # public exponent

d = 3031       # Bob leaked this!

print("Leaked private key d =", d)

print("Public exponent e    =", e)

print("Modulus n            =", n)

# Attack: compute phi(n) from e and d

# ed = 1 (mod phi(n))  -> ed - 1 = k * phi(n)

ED_minus_1 = e*d - 1

phi_candidates = []

for k in range(1, 5000):

    if ED_minus_1 % k == 0:

        phi_candidates.append(ED_minus_1 // k)

print("\nPossible phi(n) values:", phi_candidates)

# For each phi, try to factor n using:

# p + q = n - phi(n) + 1

# pq = n

for phi in phi_candidates:

    S = n - phi + 1              # p + q

    D = S*S - 4*n                # discriminant
```

```
if D >= 0:
    root = int(math.isqrt(D))
    if root * root == D:
        p = (S + root) // 2
        q = (S - root) // 2
        if p*q == n:
            print("\nRecovered factors!")
            print("phi(n) =", phi)
            print("p =", p)
            print("q =", q)
            break
```