Q1

1) Diffie-Hellman key exchange is also known as exponential key exchange. It is a method of digital encryption that uses numbers raised to specific powers to produce decryption keys on the basis of components which aren't transmitted directly. This is used to exchange the secret key between the sender and the receiver.
For example :- Credit card transaction email.

2) $q = 17 \Rightarrow n$ Alice's secret key = 4
$a = 5$ Bob's secret key = 6

$\therefore$ Public key = $a^{\text{secret key}} \bmod n$

For alice : For Bob :

$5^4 \bmod 17$ $5^6 \bmod 17$

$PK_A = 13$ $PK_B = 2$

Secret key = $PK^{\text{secret key}} \bmod n$

For Alice : For Bob :

$PK_B^{\text{secret key}} \bmod n$ $PK_A^{\text{secret key}} \bmod n$

$= 2^4 \bmod 17$ $= 13^6 \bmod 17$

$= 16$ $= 16$

$\therefore$ The secret key they exchanged is 16

Date 3/11/22
Page

Q Encryption & Decryption code for Vigenère cipher

Encryption :

```
def encrypt_cipherText(string, key):
    key = list(key)
    if len(string) == len(key):
    return(key)
else:
    for i in range(len(string) - len(key)):
        key.append(key[i % len(key)])
    return("".join(key))
```

```
def encrypt_cipherText(string, key):
    cipher_text = []
    for is in range(len(string)):
        x = ((ord(string[i]) + ord(key[i])) %. 26) + ord('A')
        cipher_text.append(chr(x))
    return("".join(cipher_text))
```

Decryption :

```
def decrypt_originalText(cipher_txt, key):
    orig_text = []
    for i in range(len(cipher_text)):
        x = ((ord(cipher_text[i]) - ord(key[i])) % 26) + ord('A')
        orig_text.append(chr(x))
    return("".join(orig_text)).
```