Tasks: 2025/10/09

Q0. WAP to print composite numbers upto 'n' number.

In [17]:
```
# composite numbers upto 'n': without a function
num = int(input("Composite No. upto: "))

if num <= 1:
    print("Please, enter non-negative number that's greater than 1 for composite numbers.")
else:
    for i in range (2, num + 1):
        for j in range (2, i):
            if (i % j) == 0:
                print(i, end = ", ") # composite numbers
                break
```

```
4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 36, 3
8, 39, 40, 42, 44, 45, 46, 48, 49, 50,
```

In [18]:
```
# composite numbers upto 'n': using a function
def composite(num):
    if num <= 1:
        print("Please, enter non-negative number that's greater than 1 for composite numbers.
    else:
        for i in range (2, num + 1):
            for j in range (2, i):
                if (i % j) == 0:
                    print(i, end = ", ") # composite numbers
                    break

num = int(input("Composite No. upto: "))
composite(num)
```

```
4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 36, 3
8, 39, 40, 42, 44, 45, 46, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68,
69, 70, 72, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 9
8, 99, 100,
```

Q1. WAP to print prime numbers till 'n' numbers.

In [11]:
```
# prime numbers upto 'n': without a function
num = int(input("Prime No. upto: "))

if num <= 1:
    print("Please, enter non-negative number that's greater than 1 for prime numbers.")
else:
    for i in range (2, num + 1):
        for j in range (2, i):
            if (i % j) == 0:
                # print(i, end = ", ") # composite numbers
                break
        else:
            print(i, end = ", ")
```

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47,
```

In [10]:
```python
# prime numbers upto 'n': using a function

def prime(num):
    if num <= 1:
        print("Please, enter non-negative number that's greater than 1 for prime numbers.")
    else:
        for i in range (2, num + 1):
            for j in range (2, i):
                if (i % j) == 0:
                    # print(i, end = ", ") # composite numbers
                    break
            else:
                print(i, end = ", ")

num = int(input("Prime No. upto: "))
prime(num)
```

```
2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 9
7,
```

Q2. WAP to print the given patterns

2.1 WAP to print the given pattern upto 'n' number.

```
x
xx
xxx
xxxx
xxxxx
```

In [138…
```python
# Without a function
x = int(input("Number: "))

for i in range (1, x + 1):
    print("x"*i)
```

```
x
xx
xxx
xxxx
xxxxx
xxxxxx
xxxxxxx
xxxxxxxx
xxxxxxxxx
xxxxxxxxxx
```

In [137…
```python
# Using a function
def pattern1(x):
    for i in range (1, x + 1):
        print("x"*i)

num = int(input("Number: "))
pattern1(num)
```

```
x
xx
xxx
xxxx
xxxxx
xxxxxx
xxxxxxx
xxxxxxxx
xxxxxxxxx
xxxxxxxxxx
```

2.2 WAP to print the given pattern upto 'n' number.

```
    x
   xx
  xxx
 xxxx
xxxxx
```

In [136…

```
# Without a function
x = int(input("Number: "))
for i in range (1, x + 1):
    print(" " * (x-i) + "x"*i)
```

```
    x
   xx
  xxx
 xxxx
xxxxx
 xxxxxx
 xxxxxxx
 xxxxxxxx
 xxxxxxxxx
xxxxxxxxxx
```

In [135…

```
# Using a function
def pattern2(x):
    for i in range (1, x + 1):
        print(" " * (x-i) + "x"*i)

num = int(input("Number: "))
pattern2(num)
```

```
    x
   xx
  xxx
 xxxx
xxxxx
 xxxxxx
 xxxxxxx
 xxxxxxxx
 xxxxxxxxx
xxxxxxxxxx
```

2.3 WAP to print the pyramid pattern of 'x' upto 'n' number.

In [134…
```
# Without a function
x = int(input("Number: "))

for i in range (1, x + 1):
    print(" "* (x-i) + "x "*i)
```

```
        x
       x x
      x x x
     x x x x
    x x x x x
   x x x x x x
  x x x x x x x
 x x x x x x x x
x x x x x x x x x
```

In [133…
```
# Using a function
def pattern3(x):
    for i in range (1, x + 1):
        print(" "* (x-i) + "x "*i)

num = int(input("Number: "))
pattern3(num)
```

```
        x
       x x
      x x x
     x x x x
    x x x x x
   x x x x x x
  x x x x x x x
 x x x x x x x x
x x x x x x x x x
```

2.4 WAP to print the inverse pyramid pattern of 'x' upto 'n' number.

In [132…
```
# Without a function
x = int(input("Number: "))

for i in range (x, 0, -1):
    print(" "* (x-i) + "x "*i)
```

```
x x x x x x x x x
 x x x x x x x x x
  x x x x x x x x
   x x x x x x x
    x x x x x x
     x x x x x
      x x x x
       x x x
        x x
         x
```

In [131…
```
# Using a function
def pattern4(x):
    for i in range (x, 0, -1):
```

```
        print(" "* (x-i) + "x "*i)

num = int(input("Number: "))
pattern4(num)
```

```
x x x x x x x x x
 x x x x x x x x x
  x x x x x x x x
   x x x x x x x
    x x x x x x
     x x x x x
      x x x x
       x x x
        x x
         x
```

2.3 WAP to print the border of 'x' with space inside upto 'n' number.

In [130…

```
# Without a function

x = int(input("Number: "))
# print(f"N: {x}")

for i in range (1, x+1):
    if i == 1:
        print("x "*x)
        continue

    if i == x:
        print("x "*x)
        break

    print(f"x " + "  " * (x-2) + "x")
```

```
x x x x x x x x x
x               x
x               x
x               x
x               x
x               x
x               x
x               x
x               x
x x x x x x x x x
```

In [129…

```
# Using a function

def pattern5(x):
    for i in range (1, x+1):
        if i == 1:
            print("x "*x)
            continue

        if i == x:
            print("x "*x)
            break

        print(f"x " + "  " * (x-2) + "x") #
```

```
num = int(input("Number: "))
pattern5(num)
```

```
x x x x x x x x x
x               x
x               x
x               x
x               x
x               x
x               x
x               x
x               x
x x x x x x x x x
```

In [128…

```
# Sister Ashira's code: tested
n = int(input("Enter: "))

for i in range(1, n + 1): # for row
    for j in range(1, n + 1): # for column
        if i == 1 or i == n or j == 1 or j == n : #check for 4 borders
            print('x', end = ' ')
        else:
            print(' ', end = ' ') # spacing
    print()
```

```
x x x x x x x x x
x               x
x               x
x               x
x               x
x               x
x               x
x               x
x               x
x x x x x x x x x
```