Assignment 3

Atul Dhiman

25901325

1. Generate all prime numbers less than 100.Print them in a spiral format (like a clockwise square spiral) of minimal size that fits all primes.

In [ ]:

```
import math
def is_prime(n):
    """Checks if a number is prime."""
    if n < 2:
        return False
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            return False
    return True

primes = [i for i in range(100) if is_prime(i)]

size = int(math.ceil(math.sqrt(len(primes))))
if size % 2 == 0:
    size += 1

spiral = [['  ' for _ in range(size)] for _ in range(size)]

x, y = 0, 0
dx, dy = 1, 0
prime_idx = 0
for _ in range(size*size):
    if 0 <= x < size and 0 <= y < size and spiral[y][x] == '  ' and prime_idx
        spiral[y][x] = str(primes[prime_idx]).zfill(2)
        prime_idx += 1

    nx, ny = x + dx, y + dy
    if not (0 <= nx < size and 0 <= ny < size and spiral[ny][nx] == '  '):
        dx, dy = -dy, dx
    x, y = x + dx, y + dy
print("Prime numbers less than 100 in a spiral format:")
for row in spiral:
    print(' '.join(row))
```

```
Prime numbers less than 100 in a spiral format:
02 03 05 07 11
53 59 61 67 13
47 89 97 71 17
43 83 79 73 19
41 37 31 29 23
```

2. You are a data analyst at a streaming platform, managing user interactions, analytics, and daily operations. Use a for loop to create a text-based graph of hourly views. For example, use * to represent every 5 views.

In [ ]:
```
hourly_views = [4, 8, 11, 14, 19, 26, 38, 49, 57, 52, 48, 49, 53, 62, 74, 88,

print("Hourly Views Analysis")
print("Key: * = 5 views\n")

for hour, views in enumerate(hourly_views):

    num_asterisks = views // 5

    graph_bar = "*" * num_asterisks

    print(f"{hour:02d}:00 | {graph_bar} (Total: {views})")
```

```
Hourly Views Analysis
Key: * = 5 views

00:00 |  (Total: 4)
01:00 | * (Total: 8)
02:00 | ** (Total: 11)
03:00 | ** (Total: 14)
04:00 | *** (Total: 19)
05:00 | ***** (Total: 26)
06:00 | ******* (Total: 38)
07:00 | ********* (Total: 49)
08:00 | *********** (Total: 57)
09:00 | ********** (Total: 52)
10:00 | ********* (Total: 48)
11:00 | ********* (Total: 49)
12:00 | ********** (Total: 53)
13:00 | ************ (Total: 62)
14:00 | ************** (Total: 74)
15:00 | ***************** (Total: 88)
16:00 | ******************** (Total: 103)
17:00 | ********************** (Total: 114)
18:00 | ************************ (Total: 120)
19:00 | ********************* (Total: 109)
20:00 | ****************** (Total: 93)
21:00 | *************** (Total: 77)
22:00 | ********* (Total: 46)
23:00 | ***** (Total: 28)
```

3. Each video watched generates revenue based on ad impressions:

- First 10 views: $0.50 per view
- Next 20 views: $0.30 per view

- Remaining views: $0.10 per view

  You are given hourly views for the day. Calculate total daily revenue.

In [ ]:

```
def calculate_daily_revenue(views_data):
    total_views = sum(views_data)
    revenue = 0.0
    if total_views <= 0:
        return 0.0
    tier1_views = min(total_views, 10)
    revenue += tier1_views * 0.50
    if total_views > 10:
        remaining_views_after_tier1 = total_views - 10
        tier2_views = min(remaining_views_after_tier1, 20)
        revenue += tier2_views * 0.30

    if total_views > 30:
        tier3_views = total_views - 30
        revenue += tier3_views * 0.10

    return revenue
hourly_views = [4, 8, 11, 14, 19, 26, 38, 49, 57, 52, 48, 49, 53, 62, 74, 88,
total_revenue = calculate_daily_revenue(hourly_views)
print(f"Total Daily Views: {sum(hourly_views)}")
print(f"Total Daily Revenue: ${total_revenue:.2f}")
```

```
Total Daily Views: 1342
Total Daily Revenue: $142.20
```

In [ ]: