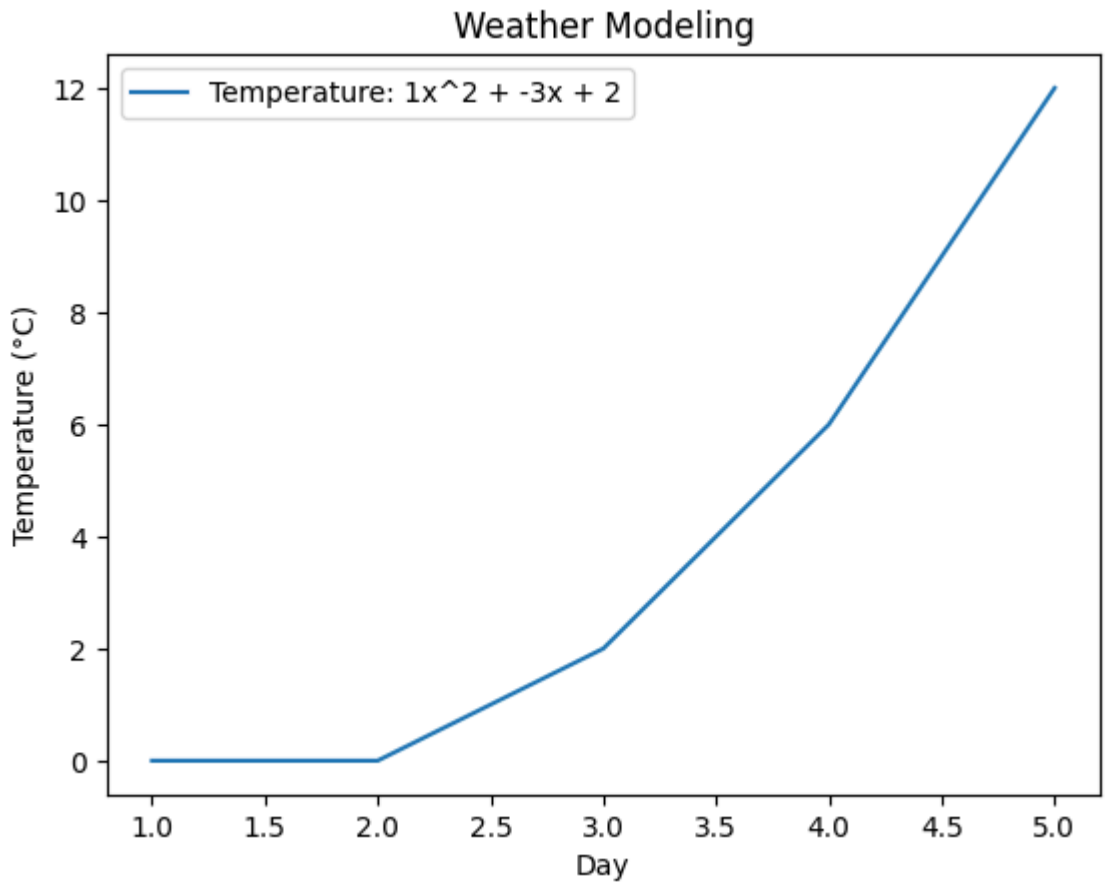
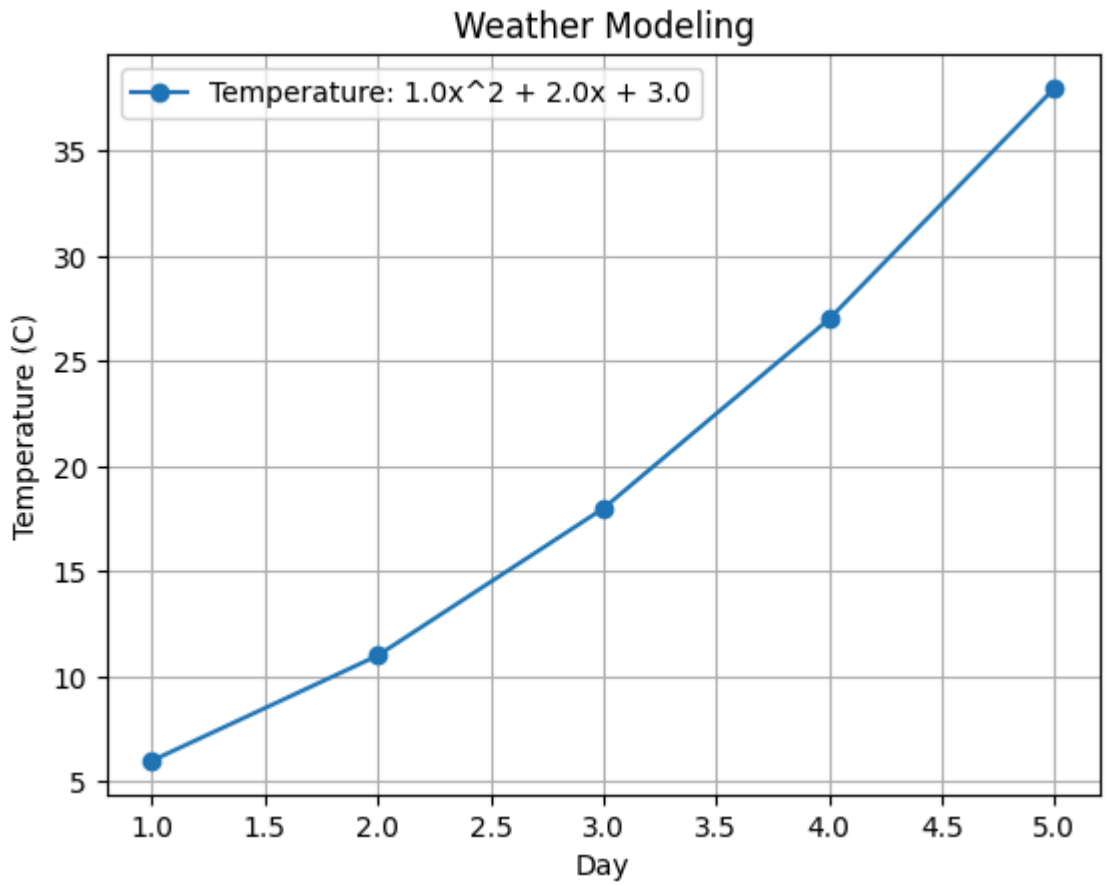
```
 import matplotlib.pyplot as plt

def quadratic_temperature_model(day, a, b, c):
    temperature = a * day**2 + b * day + c  # Quadratic equation for temperature
    return temperature

def main():
    # Fixed coefficients for the quadratic equation: ax^2 + bx + c
    a, b, c = 1, -3, 2

    # Fixed number of days to model
    num_days = int(input("Enter the number of days to model: "))

    # Lists to store day and corresponding temperature values
    days = list(range(1, num_days + 1))
    temperatures = [quadratic_temperature_model(day, a, b, c) for day in days]

    # Plotting
    plt.plot(days, temperatures, label=f'Temperature: {a}x^2 + {b}x + {c}')
    plt.title('Weather Modeling')
    plt.xlabel('Day')
    plt.ylabel('Temperature (°C)')
    plt.legend()
    plt.show()

main()
```



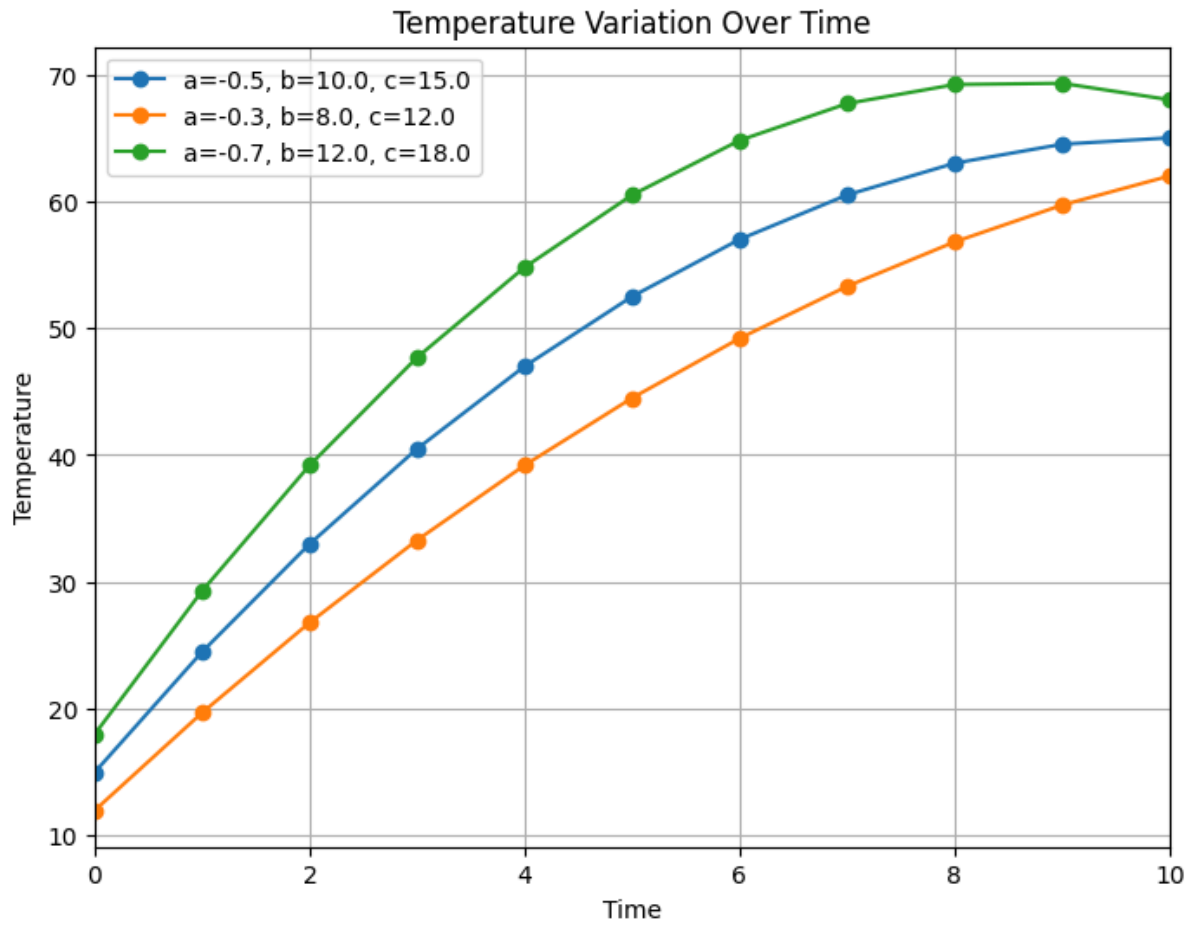

```
print("Invalid input. Please enter an integer for the number of days.")
        return

    if num_days <= 0:
        print("Number of days must be a positive integer.")
        return

    days = list(range(1, num_days + 1))
    temperatures = [quadratic_temperature_model(day, a, b, c) for day in
days]

    # Plotting
    plt.plot(days, temperatures, marker='o', label=f'Temperature: {a}x^2
+ {b}x + {c}')
    plt.title("Weather Modeling")
    plt.xlabel("Day")
    plt.ylabel("Temperature (C)")
    plt.grid(True)
    plt.legend()
    plt.show()
```

```
if __name__ == "__mimport matplotlib.pyplot as plt

def quadratic_temperature_model(day, a, b, c):
    """
    Calculates temperature using a quadratic equation: ax^2 + bx + c.

    Args:
        day (int): The day number.
        a (float): The quadratic coefficient.
        b (float): The linear coefficient.
        c (float): The constant term.

    Returns:
        float: The calculated temperature.
    """
    temperature = a * (day**2) + b * day + c
    return temperature

def main():
    """
    Main function to get user input, model temperature, and plot the
results.
    """
    # Get user input for coefficients
    try:
        a = float(input("Enter the quadratic coefficient (a): "))
        b = float(input("Enter the linear coefficient (b): "))
        c = float(input("Enter the constant term (c): "))
    except ValueError:
        print("Invalid input. Please enter numerical values for the
coefficients.")
        return

    # Get user input for the number of days to model
    try:
        num_days = int(input("Enter the number of days to model: "))
    except ValueError:ain__":
    main()
```



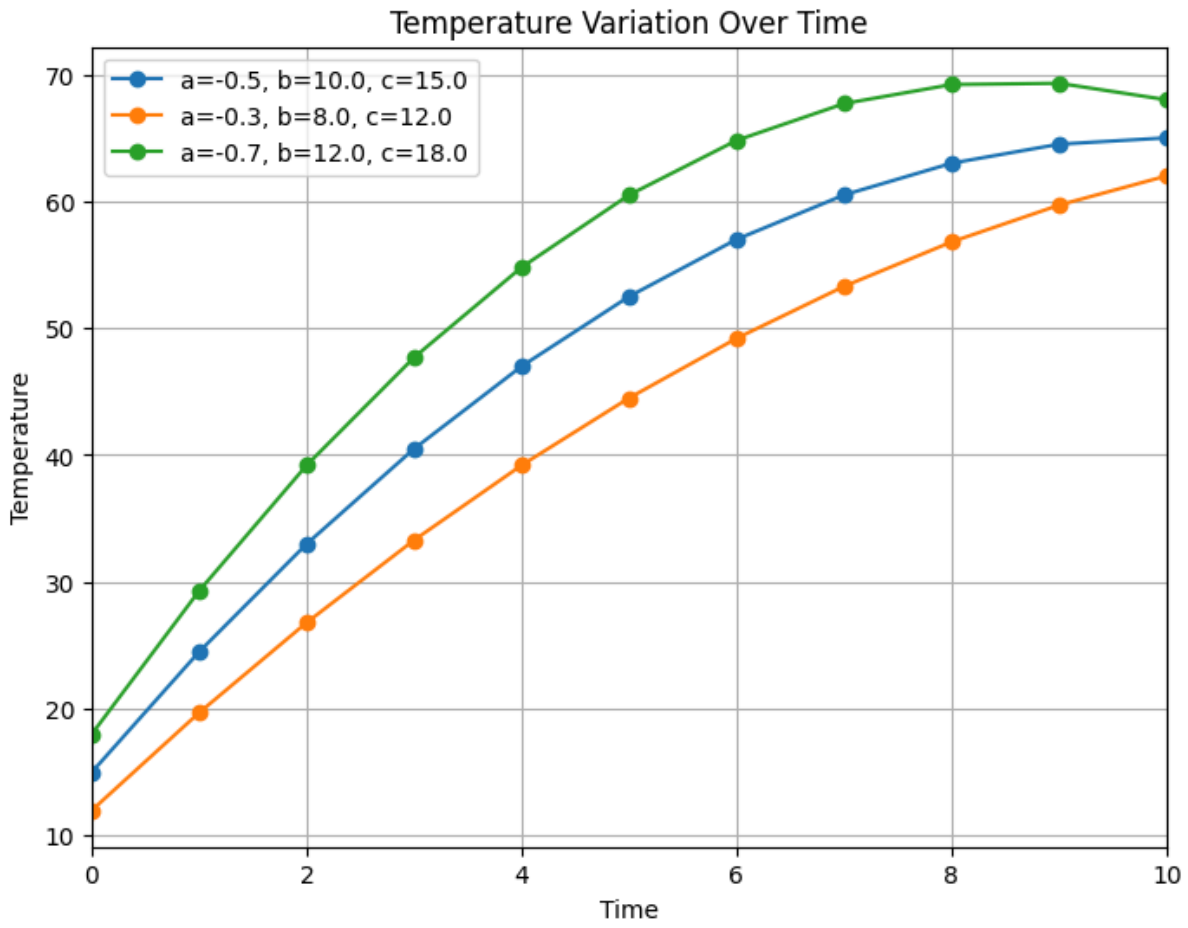

```python
import matplotlib.pyplot as plt
import pandas as pd

# Step 1: Create the CSV directly in Colab
data = {
    "a": [-0.5, -0.3, -0.7],
    "b": [10, 8, 12],
    "c": [15, 12, 18]
}
df = pd.DataFrame(data)
df.to_csv("wea2.csv", index=False)
print("✅ CSV file created: wea2.csv")

# Step 2: Define the model
def quadratic_weather_model(time, a, b, c):
    return a * (time ** 2) + b * time + c

# Step 3: Main function
def main():
```

```
    print("\nQuadratic Weather Modeling")
    print("==========================")

    file_path = 'wea2.csv'
    df = pd.read_csv(file_path)
    time_values = list(range(0, 11))

    plt.figure(figsize=(8, 6))

    for _, row in df.iterrows():
        a, b, c = row['a'], row['b'], row['c']
        temperature_values = [quadratic_weather_model(t, a, b, c) for t
in time_values]
        plt.plot(time_values, temperature_values, marker='o',
linestyle='-',
                 label=f'a={a}, b={b}, c={c}')

    plt.title('Temperature Variation Over Time')
    plt.xlabel('Time')
    plt.ylabel('Temperature')
    plt.grid(True)
    plt.xlim(0, 10)
    plt.legend()
    plt.show()

# Step 4: Run the program
main()
```




```python
import matplotlib.pyplot as plt
import pandas as pd

# Step 1: Create CSV file automatically
data = {
    "a": [-0.5, -0.3, -0.7],
    "b": [10, 8, 12],
    "c": [15, 12, 18]
}
df = pd.DataFrame(data)
df.to_csv("wea.csv", index=False)
print("✅ 'wea.csv' file created successfully!")

# Step 2: Define the model
def quadratic_weather_model(time, a, b, c):
    return a * (time ** 2) + b * time + c

# Step 3: Main plotting function
def main():
    print("\nQuadratic Weather Modeling")
```

```
    print("==========================")

    file_path = 'wea.csv'
    df = pd.read_csv(file_path)

    time_values = list(range(0, 11))
    plt.figure(figsize=(8, 6))

    for _, row in df.iterrows():
        x, y, z = row['a'], row['b'], row['c']
        temperature_values = [quadratic_weather_model(t, x, y, z) for t
in time_values]
        plt.plot(time_values, temperature_values, marker='o',
linestyle='-',
                 label=f'a={x}, b={y}, c={z}')

    plt.title('Temperature Variation Over Time')
    plt.xlabel('Time')
    plt.ylabel('Temperature')
    plt.grid(True)
    plt.xlim(0, 10)
    plt.legend()
    plt.show()

# Step 4: Run it
main()
```


Temperature Variation Over Time
a=-0.5, b=10.0, c=15.0
a=-0.3, b=8.0, c=12.0
a=-0.7, b=12.0, c=18.0
Temperature
Time
70
60
50
40
30
20
10
0
2
4
6
8
10