EXERCISES

01 Mailing Address

In [1]:

```
def display_address():
    name = "ChatGPT"
    street = "123 AI Lane"
    city = "Innovation City"
    state = "CA"
    zip_code = "90210"

    print(f"{name}")
    print(f"{street}")
    print(f"{city}, {state} {zip_code}")

# Call the function to display the address
display_address()
```

```
ChatGPT
123 AI Lane
Innovation City, CA 90210
```

02 Hello

In [2]:

```
def greet_user():
    name = input("Please enter your name: ")
    print(f"Hello, {name}!")

# Call the function to greet the user
greet_user()
```

```
Hello, Suraj!
```

03 Area of a Room

In [3]:

```
def calculate_room_area():
    width = float(input("Please enter the width of the room in meters: "))
    length = float(input("Please enter the length of the room in meters: "))
    area = width * length
    print(f"The area of the room is {area} square meters.")

# Call the function to calculate the room area
calculate_room_area()
```

```
The area of the room is 156.0 square meters.
```

04 Area of a Field

In [4]:

```
def calculate_field_area_in_acres():
    length = float(input("Please enter the length of the field in feet: "))
    width = float(input("Please enter the width of the field in feet: "))
    area_sq_ft = length * width
    area_acres = area_sq_ft / 43560
    print(f"The area of the field is {area_acres} acres.")

# Call the function to calculate the field area in acres
calculate_field_area_in_acres()
```

```
The area of the field is 0.003581267217630854 acres.
```

05 Bottle Deposits

In [5]:

```
def calculate_refund():
    # Prompt the user for the number of containers of each size
    small_containers = int(input("Enter the number of containers holding 1 litre or
    large_containers = int(input("Enter the number of containers holding more than

    # Define the deposit amounts
    deposit_small = 8
    deposit_large = 20

    # Calculate the total refund
    total_refund = (small_containers * deposit_small) + (large_containers * deposit

    # Display the refund amount
    print(f"The total refund is ₹{total_refund:.2f}")

# Call the function to calculate and display the refund
calculate_refund()
```

```
The total refund is ₹320.00
```

06 Tax and Tip

In [6]:

```
def calculate_meal_cost():
    # Prompt the user for the cost of the meal
    meal_cost = float(input("Enter the cost of the meal: ₹"))

    # Define the tax rate and tip percentage
    tax_rate = 0.05
    tip_percentage = 0.18

    # Calculate the tax, tip, and grand total
    tax_amount = meal_cost * tax_rate
    tip_amount = meal_cost * tip_percentage
    grand_total = meal_cost + tax_amount + tip_amount
    # Display the tax, tip, and grand total amounts
    print(f"Tax amount: ₹{tax_amount:.2f}")
    print(f"Tip amount: ₹{tip_amount:.2f}")
    print(f"Grand total: ₹{grand_total:.2f}")

# Call the function to calculate and display the meal cost
calculate_meal_cost()
```

```
Tax amount: ₹6.00
Tip amount: ₹21.60
Grand total: ₹147.60
```

07 Sum of the First n Positive Integers

In [7]:

```
def sum_of_integers():
    # Prompt the user for a positive integer
    n = int(input("Enter a positive integer: "))

    # Ensure the input is a positive integer
    if n <= 0:
        print("Please enter a positive integer.")
        return

    # Calculate the sum using the formula
    sum_n = n * (n + 1) // 2

    # Display the sum
    print(f"The sum of all integers from 1 to {n} is {sum_n}")

# Call the function to calculate and display the sum
sum_of_integers()
```

```
The sum of all integers from 1 to 12 is 78
```

08 Tax and Tip

In [8]:

```
def calculate_total_weight():
    # Prompt the user for the number of widgets and gizmos
    num_widgets = int(input("Enter the number of widgets: "))
    num_gizmos = int(input("Enter the number of gizmos: "))

    # Define the weights of the widgets and gizmos in grams
    weight_widget = 75
    weight_gizmo = 112

    # Calculate the total weight
    total_weight = (num_widgets * weight_widget) + (num_gizmos * weight_gizmo)

    # Display the total weight
    print(f"The total weight of the parts is {total_weight} grams.")
```

```
# Call the function to calculate and display the total weight
calculate_total_weight()
```

```
The total weight of the parts is 10344 grams.
```

09 Compound Interest

In [9]:

```
def calculate_savings():
    # Prompt the user for the initial deposit amount
    initial_deposit = float(input("Enter the amount of money deposited: ₹"))

    # Define the annual interest rate
    interest_rate = 0.04

    # Calculate the amount after each year
    amount_year_1 = initial_deposit * (1 + interest_rate)
    amount_year_2 = amount_year_1 * (1 + interest_rate)
    amount_year_3 = amount_year_2 * (1 + interest_rate)

    # Display the amounts rounded to 2 decimal places
    print(f"Amount in the savings account after 1 year: ₹{amount_year_1:.2f}")
    print(f"Amount in the savings account after 2 years: ₹{amount_year_2:.2f}")
    print(f"Amount in the savings account after 3 years: ₹{amount_year_3:.2f}")

# Call the function to calculate and display the savings
calculate_savings()
```

```
Amount in the savings account after 1 year: ₹2080.00
Amount in the savings account after 2 years: ₹2163.20
Amount in the savings account after 3 years: ₹2249.73
```

10 Arithmetic

In [13]:

```
import math

def perform_operations():
    # Prompt the user for input values
    a = float(input("Enter the first number (a): "))
    b = float(input("Enter the second number (b): "))

    # Perform the calculations
    sum_ab = a + b
    difference_ab = a - b
    product_ab = a * b
    quotient_ab = a / b if b != 0 else 'undefined'  # Handle division by zero
    remainder_ab = a % b if b != 0 else 'undefined'  # Handle division by zero
    log_a = math.log10(a) if a > 0 else 'undefined'  # Handle log of non-positive n
    power_ab = a ** b

    # Display the results
    print(f"The sum of {a} and {b} is {sum_ab}")
    print(f"The difference when {b} is subtracted from {a} is {difference_ab}")
    print(f"The product of {a} and {b} is {product_ab}")
    print(f"The quotient when {a} is divided by {b} is {quotient_ab}")
    print(f"The remainder when {a} is divided by {b} is {remainder_ab}")
    print(f"The result of log_10({a}) is {log_a}")
    print(f"The result of {a}^{b} is {power_ab}")

# Call the function to perform the operations
perform_operations()
```

```
The sum of 12.0 and 13.0 is 25.0
The difference when 13.0 is subtracted from 12.0 is -1.0
The product of 12.0 and 13.0 is 156.0
The quotient when 12.0 is divided by 13.0 is 0.9230769230769231
The remainder when 12.0 is divided by 13.0 is 12.0
The result of log_10(12.0) is 1.0791812460476249
The result of 12.0^13.0 is 106993205379072.0
```

11 Fuel Efficiencey

In [14]:

```
def mpg_to_l_per_100km():
    # Prompt the user for the fuel efficiency in MPG
    mpg = float(input("Enter the fuel efficiency in miles per gallon (MPG): "))

    # Convert MPG to L/100 km using the conversion factor
    l_per_100km = 235.215 / mpg

    # Display the equivalent fuel efficiency in L/100 km
    print(f"The equivalent fuel efficiency is {l_per_100km:.2f} L/100 km.")

# Call the function to perform the conversion
mpg_to_l_per_100km()
```

```
The equivalent fuel efficiency is 1.96 L/100 km.
```

12 Distance Between Two Points on Earth

In [15]:

```
import math

def haversine_distance(lat1, lon1, lat2, lon2):
    # Radius of the Earth in kilometers
    R = 6371.0

    # Convert latitude and longitude from degrees to radians
    lat1_rad = math.radians(lat1)
    lon1_rad = math.radians(lon1)
    lat2_rad = math.radians(lat2)
    lon2_rad = math.radians(lon2)

    # Haversine formula
    dlon = lon2_rad - lon1_rad
    dlat = lat2_rad - lat1_rad
    a = math.sin(dlat / 2)**2 + math.cos(lat1_rad) * math.cos(lat2_rad) * math.sin(
    c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
    distance = R * c
    return distance

# Example usage:
if __name__ == "__main__":
    # Coordinates for New York City (40.7128° N, 74.0060° W)
    lat1 = 40.7128
    lon1 = -74.0060

    # Coordinates for Los Angeles (34.0522° N, 118.2437° W)
    lat2 = 34.0522
    lon2 = -118.2437

    # Calculate the distance between New York City and Los Angeles
    distance = haversine_distance(lat1, lon1, lat2, lon2)
    print(f"The distance between New York City and Los Angeles is {distance:.2f} ki
```

```
The distance between New York City and Los Angeles is 3935.75 kilometers.
```

13 Making Change

In [16]:

```
def calculate_change():
    # Prompt the user for the amount in cents
    cents = int(input("Enter the amount in cents: "))

    # Initialize coin denominations in cents
    toonies_value = 200
    loonies_value = 100
    quarters_value = 25
    dimes_value = 10
    nickels_value = 5
    pennies_value = 1

    # Calculate the number of each coin denomination
    toonies = cents // toonies_value
    cents %= toonies_value

    loonies = cents // loonies_value
    cents %= loonies_value

    quarters = cents // quarters_value
    cents %= quarters_value

    dimes = cents // dimes_value
    cents %= dimes_value

    nickels = cents // nickels_value
    cents %= nickels_value

    pennies = cents

    # Display the number of each coin denomination
    print("Change to be given:")
    if toonies > 0:
        print(f"Toonies: {toonies}")
```

```python
    if loonies > 0:
        print(f"Loonies: {loonies}")
    if quarters > 0:
        print(f"Quarters: {quarters}")
    if dimes > 0:
        print(f"Dimes: {dimes}")
    if nickels > 0:
        print(f"Nickels: {nickels}")
    if pennies > 0:
        print(f"Pennies: {pennies}")

# Call the function to calculate and display the change
calculate_change()
```

```
Change to be given:
Loonies: 1
Dimes: 2
```

14 Height Units

In [17]:

```python
def feet_inches_to_cm():
    # Prompt the user for feet and inches
    feet = int(input("Enter the number of feet: "))
    inches = int(input("Enter the number of inches: "))

    # Convert feet and inches to centimeters
    total_inches = feet * 12 + inches
    cm = total_inches * 2.54

    # Display the equivalent height in centimeters
    print(f"The equivalent height is {cm} centimeters.")

# Call the function to perform the conversion
feet_inches_to_cm()
```

```
The equivalent height is 3779.52 centimeters.
```

15 Distance Units

In [18]:

```python
def convert_feet():
    # Prompt the user for the measurement in feet
    feet = float(input("Enter the measurement in feet: "))

    # Conversion factors
    inches_per_foot = 12
    yards_per_foot = 1 / 3  # 1 yard = 3 feet
    miles_per_foot = 1 / 5280  # 1 mile = 5280 feet

    # Convert feet to inches, yards, and miles
    inches = feet * inches_per_foot
    yards = feet * yards_per_foot
    miles = feet * miles_per_foot
    # Display the equivalent distances
    print(f"The equivalent distance in inches is {inches} inches.")
    print(f"The equivalent distance in yards is {yards} yards.")
    print(f"The equivalent distance in miles is {miles} miles.")

# Call the function to perform the conversion
convert_feet()
```

```
The equivalent distance in inches is 1476.0 inches.
The equivalent distance in yards is 41.0 yards.
The equivalent distance in miles is 0.023295454545454546 miles.
```

16 Area and Volumne

In [19]:

```
import math

def calculate_circle_area_and_sphere_volume():
    # Prompt the user for the radius
    r = float(input("Enter the radius of the circle and sphere (in meters): "))

    # Calculate the area of the circle
    area_circle = math.pi * r**2

    # Calculate the volume of the sphere
    volume_sphere = (4/3) * math.pi * r**3

    # Display the results
    print(f"The area of the circle with radius {r} meters is {area_circle:.2f} squa
    print(f"The volume of the sphere with radius {r} meters is {volume_sphere:.2f}

# Call the function to perform the calculations
calculate_circle_area_and_sphere_volume()
```

```
The area of the circle with radius 12.0 meters is 452.39 square meters.
The volume of the sphere with radius 12.0 meters is 7238.23 cubic meters.
```

17 Volume of a Cylinder

In [20]:

```
import math

def calculate_cylinder_volume():
    # Prompt the user for the radius and height of the cylinder
    radius = float(input("Enter the radius of the cylinder (in meters): "))
    height = float(input("Enter the height of the cylinder (in meters): "))

    # Calculate the area of the circular base
    base_area = math.pi * radius**2

    # Calculate the volume of the cylinder
    volume = base_area * height

    # Display the volume rounded to one decimal place
    print(f"The volume of the cylinder is {volume:.1f} cubic meters.")
```

```
# Call the function to calculate and display the cylinder volume
calculate_cylinder_volume()
```

```
The volume of the cylinder is 5881.1 cubic meters.
```

18 Free Fall

In [21]:

```
import math

def calculate_final_speed():
    # Prompt the user for the height from which the object is dropped
    height = float(input("Enter the height from which the object is dropped (in met

    # Acceleration due to gravity in meters per second squared
    g = 9.8

    # Calculate the final speed using the formula v = sqrt(2gh)
    final_speed = math.sqrt(2 * g * height)

    # Display the final speed rounded to two decimal places
    print(f"The final speed of the object when it hits the ground is {final_speed:.

# Call the function to calculate and display the final speed
calculate_final_speed()
```

```
The final speed of the object when it hits the ground is 15.34 m/s.
```

19 Heat Capacity

In [23]:

```
def calculate_energy_and_cost():
    # Constants
    specific_heat_water = 4.186  # J/g°C
    joules_to_kwh = 1 / 3600000  # Conversion factor from Joules to kWh
    electricity_cost_per_kwh = 8.9  # ₹ per kWh

    # Input from user
    mass_water = float(input("Enter the mass of water (in grams or milliliters): "))
    temp_change = float(input("Enter the temperature change (in degrees Celsius): "

    # Calculate the energy required in Joules
    energy_joules = mass_water * specific_heat_water * temp_change

    # Convert energy from Joules to kilowatt-hours (kWh)
    energy_kwh = energy_joules * joules_to_kwh

    # Calculate the cost of heating the water
    cost = energy_kwh * electricity_cost_per_kwh

    # Display the results
    print(f"The total energy required to heat {mass_water} grams of water by {temp_
    print(f"This is equivalent to {energy_kwh:.6f} kWh.")
    print(f"The cost of heating the water is ₹{cost:.2f}.")
```

```
# Call the function to calculate energy and cost
calculate_energy_and_cost()
```

```
The total energy required to heat 10.0 grams of water by 200.0 degrees Celsius is
8372.00 Joules.
This is equivalent to 0.002326 kWh.
The cost of heating the water is ₹0.02.
```

20 Area of a Triangle

In [24]:

```
def calculate_triangle_area():
    # Prompt the user for the base length and height of the triangle
    b = float(input("Enter the base length of the triangle: "))
    h = float(input("Enter the height of the triangle: "))

    # Calculate the area of the triangle using the formula: Area = 0.5 * base * hei
    area = 0.5 * b * h

    # Display the calculated area
    print(f"The area of the triangle with base length {b} and height {h} is {area}

# Call the function to calculate and display the triangle area
calculate_triangle_area()
```

```
The area of the triangle with base length 12.0 and height 13.0 is 78.0 square unit
s.
```

21. Area of a Triangle (Again)

In [25]:

```
import math

def calculate_triangle_area_with_sides():
    # Prompt the user for the lengths of the sides of the triangle
    s1 = float(input("Enter the length of the first side of the triangle: "))
    s2 = float(input("Enter the length of the second side of the triangle: "))
    s3 = float(input("Enter the length of the third side of the triangle: "))

    # Calculate the semi-perimeter
    s = (s1 + s2 + s3) / 2

    # Calculate the area using Heron's formula
    area = math.sqrt(s * (s - s1) * (s - s2) * (s - s3))

    # Display the calculated area
    print(f"The area of the triangle with sides {s1}, {s2}, {s3} is {area} square u

# Call the function to calculate and display the triangle area
calculate_triangle_area_with_sides()
```

```
The area of the triangle with sides 12.0, 13.0, 14.0 is 72.30793524918272 square u
nits.
```

22.Ideal Gas Law

In [28]:

```
def calculate_moles_of_gas():
    # Constants
    R = 8.314  # Ideal gas constant in J/mol·K

    # Prompt the user for input
    pressure = float(input("Enter the pressure of the gas in Pascals (Pa): "))
    volume = float(input("Enter the volume of the gas in liters (L): "))
    temperature_celsius = float(input("Enter the temperature of the gas in Celsius

    # Convert Celsius to Kelvin
    temperature_kelvin = temperature_celsius + 273.15

    # Calculate the amount of gas in moles using the ideal gas law
    n = (pressure * volume) / (R * temperature_kelvin)

    # Display the calculated amount of gas in moles
    print(f"The amount of gas in the SCUBA tank is approximately {n:.2f} moles.")

# Call the function to calculate and display the moles of gas
calculate_moles_of_gas()
```

```
The amount of gas in the SCUBA tank is approximately 5284.09 moles.
```

23. Area of a Regular Polygon

In [29]:

```
import math

def calculate_regular_polygon_area():
    # Prompt the user for input
    s = float(input("Enter the length of a side of the regular polygon: "))
    n = int(input("Enter the number of sides of the regular polygon: "))

    # Calculate the area of the regular polygon using the formula
    numerator = n * s**2
    denominator = 4 * math.tan(math.pi / n)
    area = numerator / denominator

    # Display the calculated area
    print(f"The area of the regular polygon with side length {s} and {n} sides is {

# Call the function to calculate and display the area of the regular polygon
calculate_regular_polygon_area()
```

```
The area of the regular polygon with side length 12.0 and 20 sides is 4545.90 squa
re units.
```

24. Units of Time

In [30]:

```python
def calculate_total_seconds():
    # Prompt the user for input
    days = int(input("Enter number of days: "))
    hours = int(input("Enter number of hours: "))
    minutes = int(input("Enter number of minutes: "))
    seconds = int(input("Enter number of seconds: "))

    # Calculate total seconds
    total_seconds = (days * 24 * 60 * 60) + (hours * 60 * 60) + (minutes * 60) + se

    # Display the total seconds
    print(f"The total duration of {days} days, {hours} hours, {minutes} minutes, an

# Call the function to calculate and display total seconds
calculate_total_seconds()
```

```
The total duration of 12 days, 12 hours, 12 minutes, and 12 seconds is 1080732 sec
onds.
```

25. Units of Time (Again)

In [32]:

```python
def convert_seconds_to_time():
    # Prompt the user for input
    seconds = int(input("Enter number of seconds: "))

    # Calculate days, hours, minutes, and seconds
    minutes, seconds = divmod(seconds, 60)
    hours, minutes = divmod(minutes, 60)
    days, hours = divmod(hours, 24)

    # Format the time components with zero-padding
    formatted_time = f"{days}:{hours:02}:{minutes:02}:{seconds:02}"

    # Display the formatted time
    print(f"The equivalent time for {seconds} seconds is {formatted_time}.")

# Call the function to convert seconds to formatted time
convert_seconds_to_time()
```

```
The equivalent time for 40 seconds is 1:11:06:40.
```

26. Current Time

In [33]:

```python
import time

def display_current_time():
    # Get current time in human-readable format
    current_time = time.asctime()

    # Display the current time and date
    print(f"Current time and date: {current_time}")
```

```python
# Call the function to display the current time and date
display_current_time()
```

```
Current time and date: Tue Jun 25 00:46:06 2024
```

27 When is Easter?

In [35]:

```python
def compute_easter_date(year):
    # Apply the Anonymous Gregorian Computus algorithm
    a = year % 19
    b = year // 100
    c = year % 100
    d = b // 4
    e = b % 4
    f = (b + 8) // 25
    g = (b - f + 1) // 3
    h = (19 * a + b - d - g + 15) % 30
    i = c // 4
    k = c % 4
    l = (32 + 2 * e + 2 * i - h - k) % 7
    m = (a + 11 * h + 22 * l) // 451
    month = (h + l - 7 * m + 114) // 31
    day = ((h + l - 7 * m + 114) % 31) + 1

    # Determine the month and day of Easter
    if month == 3:
        month_name = "March"
    else:
        month_name = "April"

    # Return the formatted date of Easter
    return f"Easter in {year} is on {month_name} {day}"

def main():
    # Prompt the user for the year
    year = int(input("Enter a year to calculate the date of Easter: "))

    # Calculate and display the date of Easter
    easter_date = compute_easter_date(year)
    print(easter_date)

# Run the main function
main()
```

```
Easter in 2024 is on March 31
```

28 Body Mass Index

In [36]:

```python
def calculate_bmi():
    print("Choose your preferred units:")
    print("1. Inches and pounds")
    print("2. Meters and kilograms")
    choice = input("Enter your choice (1 or 2): ")
    if choice == '1':
        # Calculate BMI using inches and pounds
        height = float(input("Enter your height in inches: "))
        weight = float(input("Enter your weight in pounds: "))

        bmi = (weight * 703) / (height ** 2)
    elif choice == '2':
        # Calculate BMI using meters and kilograms
        height = float(input("Enter your height in meters: "))
        weight = float(input("Enter your weight in kilograms: "))

        bmi = weight / (height ** 2)
    else:
        print("Invalid choice. Please enter 1 or 2.")
        return

    # Display the calculated BMI
    print(f"Your BMI is: {bmi:.2f}")

# Call the function to calculate BMI
calculate_bmi()
```

```
Choose your preferred units:
1. Inches and pounds
2. Meters and kilograms

Your BMI is: 3.25
```

29 Wind Chill

In [37]:

```
def calculate_wind_chill():
    # Prompt user for input
    temperature = float(input("Enter the air temperature in degrees Celsius: "))
    wind_speed = float(input("Enter the wind speed in kilometers per hour: "))

    # Check if conditions are valid for computing wind chill index
    if temperature > 10.0:
        print("Wind chill index is not applicable for temperatures above 10°C.")
        return
    if wind_speed <= 4.8:
        print("Wind chill index is not applicable for wind speeds at or below 4.8 k
        return

    # Calculate wind chill index using the formula
    wind_chill_index = 13.12 + 0.6215 * temperature - 11.37 * (wind_speed ** 0.16)

    # Round the wind chill index to the nearest integer
    wind_chill_index = round(wind_chill_index)

    # Display the result
    print(f"The wind chill index is: {wind_chill_index}")

# Call the function to calculate wind chill index
calculate_wind_chill()
```

```
Wind chill index is not applicable for temperatures above 10°C.
```

30 Celsius to Fahreheit and Kelvin

In [38]:

```
def convert_temperature():
    # Read temperature in degrees Celsius from user
    celsius = float(input("Enter the temperature in degrees Celsius: "))

    # Convert Celsius to Fahrenheit
    fahrenheit = (celsius * 9/5) + 32

    # Convert Celsius to Kelvin
    kelvin = celsius + 273.15

    # Display the results
    print(f"Temperature in Fahrenheit: {fahrenheit:.2f} °F")
    print(f"Temperature in Kelvin: {kelvin:.2f} K")

# Call the function to convert temperature
convert_temperature()
```

```
Temperature in Fahrenheit: 248.00 °F
Temperature in Kelvin: 393.15 K
```

31. Unit of Pressure

In [39]:

```
def convert_pressure():
    # Read pressure in kilopascals from user
    kpa = float(input("Enter the pressure in kilopascals (kPa): "))

    # Convert kPa to psi
    psi = kpa / 6.89476

    # Convert kPa to mmHg
    mmhg = kpa * 7.50062

    # Convert kPa to atm
    atm = kpa * 0.00986923

    # Display the results
    print(f"Pressure in pounds per square inch (psi): {psi:.2f} psi")
    print(f"Pressure in millimeters of mercury (mmHg): {mmhg:.2f} mmHg")
    print(f"Pressure in atmospheres (atm): {atm:.2f} atm")

# Call the function to convert pressure
convert_pressure()
```

```
Pressure in pounds per square inch (psi): 17.40 psi
Pressure in millimeters of mercury (mmHg): 900.07 mmHg
Pressure in atmospheres (atm): 1.18 atm
```

Exercise 32: Sum of the Digits in an Integer

In [40]:

```python
def sum_of_digits():
    # Read a four-digit integer from the user
    number = input("Enter a four-digit integer: ")

    # Validate input to ensure it's a four-digit integer
    if not number.isdigit() or len(number) != 4:
        print("Please enter a valid four-digit integer.")
        return

    # Convert input string to integers
    digit1 = int(number[0])
    digit2 = int(number[1])
    digit3 = int(number[2])
    digit4 = int(number[3])

    # Calculate the sum of digits
    sum_digits = digit1 + digit2 + digit3 + digit4

    # Display the result
    print(f"The sum of digits {digit1} + {digit2} + {digit3} + {digit4} = {sum_digi

# Call the function to calculate sum of digits
sum_of_digits()
```

```
The sum of digits 1 + 2 + 3 + 4 = 10
```

33 Sort 3 Integers

In [41]:

```python
def sort_three_integers():
    # Read three integers from the user
    num1 = int(input("Enter the first integer: "))
    num2 = int(input("Enter the second integer: "))
    num3 = int(input("Enter the third integer: "))

    # Find the smallest, middle, and largest values
    smallest = min(num1, num2, num3)
    largest = max(num1, num2, num3)
    middle = num1 + num2 + num3 - smallest - largest

    # Display the integers in sorted order
    print(f"The integers in sorted order are: {smallest}, {middle}, {largest}")

# Call the function to sort three integers
sort_three_integers()
```

```
The integers in sorted order are: 12, 13, 14
```

34 Day Old Bread

In [42]:

```python
def calculate_bread_price():
    # Constants
    regular_price = 250  # Regular price of bread in Rs.
    discount_rate = 0.6  # Discount rate for day-old bread

    # Read the number of loaves of day-old bread from the user
    num_loaves = int(input("Enter the number of loaves of day-old bread: "))

    # Calculate regular price, discount, and total price
    regular_total = num_loaves * regular_price
    discount_amount = regular_total * discount_rate
    discounted_total = regular_total - discount_amount

    # Display the results
    print(f"Regular price: Rs. {regular_total:.2f}")
    print(f"Discount (60% off): Rs. {discount_amount:.2f}")
    print(f"Total price after discount: Rs. {discounted_total:.2f}")

# Call the function to calculate and display bread prices
calculate_bread_price()
```

```
Regular price: Rs. 3000.00
Discount (60% off): Rs. 1800.00
Total price after discount: Rs. 1200.00
```

END