

1. Suppose you are working on a dataset of medical records that includes patients' age, blood pressure, and cholesterol levels. You need to perform a clustering analysis to identify groups of patients with similar characteristics, but the large range of values in the blood pressure and cholesterol levels can lead to numerical instability and slow computation times. To address this issue, you choose to divide the blood pressure values by 64 and the cholesterol values by 256.

**Sample Input**

Age = 25

Blood pressure = 130

Cholesterol = 220

Write a Python code to display the scaled values without using any arithmetic operators.

### **Python Code:**

```
def scale_values(age, blood_pressure, cholesterol):
    blood_pressure_factor = 64
    cholesterol_factor = 256
    scaled_blood_pressure = blood_pressure >> (blood_pressure_factor.bit_length() - 1)
    scaled_cholesterol = cholesterol >> (cholesterol_factor.bit_length() - 1)
    return age, scaled_blood_pressure, scaled_cholesterol

age = 25
blood_pressure = 130
cholesterol = 220
scaled_age, scaled_blood_pressure, scaled_cholesterol = scale_values(age,
blood_pressure, cholesterol)
print(f"Scaled Age: {scaled_age}")
print(f"Scaled Blood Pressure: {scaled_blood_pressure}")
print(f"Scaled Cholesterol: {scaled_cholesterol}")
```

2. Write a Python code to initialize a number and display true if the number is even; display false if the number is odd. Write the code without using conditional and/or operators.

### **Python Code:**

```
def even_or_odd(number):
    result = number & 1      # Use bitwise AND operation to check the least significant bit
    return result == 0

number_to_check = 8
result = even_or_odd(number_to_check)
print(f"The number {number_to_check} is even: {result}")
```

**3. Write a Python code to extract the red, green, and blue values from a given pixel without using bitwise operations.**

**Sample Input**

**pixel = 1922241 or**

**pixel = 0x 1D54C1**

### **Python Code:**

```
def extract_rgb(pixel):
    blue = pixel // 256**2
    green = (pixel // 256) % 256
    red = pixel % 256
    return red, green, blue

pixel1 = 1922241
pixel2 = 0x1D54C1
red1, green1, blue1 = extract_rgb(pixel1)
red2, green2, blue2 = extract_rgb(pixel2)
print(f"Pixel 1 - Red: {red1}, Green: {green1}, Blue: {blue1}")
print(f"Pixel 2 - Red: {red2}, Green: {green2}, Blue: {blue2}")
```

Hands on practice tasks: Basic python code

Submitted by: Md. Iquball Hossain

NACTAR MLDS Batch : 03 , Roll: 12

Designation: Junior Software Engineer  
(Python Flask and PostgreSQL Database)

Server-side Backend Developer

Email: Iquball.m360ict@gmail.com

M360 ICT Software Company , Banani, Dhaka