

IP ADDRESS MASK IDENTIFICATION

CODE :

```
def convert_to_binary(ip_address):
    octets = ip_address.split('.')
    binary_ip = [format(int(octet), '08b') for octet in octets]
    return '.'.join(binary_ip)

def convert_to_decimal(binary_subnet_mask):
    decimal_subnet_mask = '.'.join([str(int(subnet, 2)) for subnet in
    binary_subnet_mask.split('.')])
    return decimal_subnet_mask

def calculate_subnet_mask(subnet_mask):
    subnet_bits = int(subnet_mask)
    if subnet_bits < 0 or subnet_bits > 32:
        return "Invalid Subnet Mask"
    subnet_binary = '1' * subnet_bits + '0' * (32 - subnet_bits)
    subnet_binary_str = '.'.join([subnet_binary[i:i + 8] for i in range(0, 32,
8)])
    subnet_binary_str.replace('.', '') # Remove the periods
    subnet_decimal_str = convert_to_decimal(subnet_binary_str)
    return subnet_binary_str, subnet_decimal_str

10
def is_classful_address(ip_address):
    octets = ip_address.split('.')
    if len(octets) != 4:
        return False
    for octet in octets:
        if not octet.isdigit() or int(octet) < 0 or int(octet) > 255:
            return False
    first_octet = int(octets[0])
    if 1 <= first_octet <= 126:
        return "Class A"
    elif 128 <= first_octet <= 191:
        return "Class B"
    elif 192 <= first_octet <= 223:
        return "Class C"
    elif 224 <= first_octet <= 239:
        return "Class D (Multicast)"
    elif 240 <= first_octet <= 255:
        return "Class E (Experimental)"
    else:
```

```

        return "Invalid classful IP"

def is_classless_address(ip_address, subnet_mask):
    # existing function remains the same'
    return "Classless Addressing"

ip_choice = input("Enter 'C' for Classful Addressing and 'CL' for Classless Addressing: ")
print()
if ip_choice.upper() == 'C':
    user_ip = input("Enter the IP address: ")
    result = is_classful_address(user_ip)
    print(f"The IP address {user_ip} belongs to {result} based on classful addressing.")
    print()
elif ip_choice.upper() == 'CL':
    user_input = input("Enter the IP address and Subnet Mask (separated by slash): ")
    user_ip, subnet_mask = user_input.split('/')
    print()
    # Convert IP address to binary
    binary_ip = convert_to_binary(user_ip)
    print(f"The IP address {user_ip} in binary is {binary_ip}")
    print()
    # Calculate subnet mask
    subnet_binary_str, subnet_decimal_str = calculate_subnet_mask(subnet_mask)
    print(f"The Subnet Mask for {subnet_mask} is {subnet_binary_str} in binary")
    print()
    print(f"The subnet mask for {subnet_mask} is {subnet_decimal_str} in decimal.")
    print()
    result = is_classless_address(user_ip, subnet_mask)
    print(f"The IP address {user_ip} with Subnet Mask {subnet_mask} belongs to {result} based on classless addressing.")
else:
    print("Invalid choice. Please enter either 'C' or 'CL'.")

```