Exercises related to types of Function arguments

1. **Positional Arguments:**
   a. Write a Python function called `calculate_bill` that takes three positional arguments: `item_price`, `quantity`, and `tax_rate`. Calculate the total bill amount including tax and return it.
   b. Create a program that prompts the user to enter the lengths of three sides of a triangle as positional arguments to a function named `calculate_triangle_area`. Calculate the area of the triangle using Heron's formula and return the result.
   c. Implement a function called `display_student_info` that accepts three positional arguments: `name`, `age`, and `grade`. Print out the information in a formatted manner.

2. **Keyword Arguments:**
   a. Write a Python function named `print_person_info` that takes three keyword arguments: `name`, `age`, and `city`. Print out the person's information using these arguments.
   b. Create a program that defines a function called `calculate_total_cost` which accepts keyword arguments for `price`, `quantity`, and `discount`. Calculate the total cost after applying the discount and return it.
   c. Implement a function called `create_car` that takes keyword arguments like `make`, `model`, `year`, and `color`. Return a dictionary containing these details of the car.

3. **Default Arguments:**
   a. Define a function called `greet_user` that takes a default argument `name` with a default value of "Guest". The function should print a personalized greeting message.
   b. Write a program that defines a function `calculate_power` with two arguments: `base` and `exponent`, with a default value of 2 for `exponent`. Calculate and return the result of raising `base` to the power of `exponent`.
   c. Implement a function called `create_email` that takes default arguments for `subject` (default: "Hello") and `body` (default: "How are you?"). The function should return a formatted email message.

4. **Arbitrary Arguments:**
   a. Create a function named `calculate_sum` that accepts an arbitrary number of arguments and returns the sum of all the arguments.
   b. Define a function called `find_max` that takes in an arbitrary number of integers as arguments and returns the maximum value.
   c. Write a program that defines a function `concat_strings` to concatenate an arbitrary number of strings passed as arguments.

5. **Arbitrary Keyword Arguments:**
   a. Write a Python function called `print_student_details` that accepts arbitrary keyword arguments representing student information such as name, age, grade, and prints them out.
   b. Implement a function named `calculate_product_price` that accepts arbitrary keyword arguments for item prices and quantities, calculates the total cost for each item, and returns a dictionary with item names as keys and total costs as values.

c. Create a program that defines a function called `display_info` which accepts arbitrary keyword arguments representing person's details like name, age, city, and prints them out.

6. **Mix of All Arguments:**
   a. Define a function named `process_order` that takes a mix of positional arguments (`product_name`, `quantity`) and keyword arguments (`price`, `discount`) and calculates the total cost of the order.
   b. Write a program that defines a function `send_email` which takes a mix of positional and keyword arguments such as recipient email, subject, body, and additional options.
   c. Implement a function called `create_account` that accepts a mix of positional arguments (`username`, `password`) and arbitrary keyword arguments (`email`, `phone_number`) to create a user account.