

1. Define a Function with Keyword Arguments:

Question: Define a function called `greet` that takes in two keyword arguments, `name` and `msg`. The function should return a greeting in the format `msg name!`.

By default, `name` should be `"Guest"` and `msg` should be `"Hello"`.

2. Swapping Two Numbers Using Keyword Arguments:

Question: Define a function called `swap` that takes in two keyword arguments, `a` and `b`. The function should return the numbers swapped.

3. Calculating Area of a Rectangle:

Question: Define a function called `rectangle_area` that calculates the area of a rectangle. The function should take two keyword arguments: `length` and `width`.

4. Using `*args` to find the Average:

Question: Define a function called `average` that takes any number of positional arguments and returns their average.

5. Using `**kwargs` to Build a Profile:

Question: Define a function called `build_profile` that takes any number of keyword arguments. The function should return a dictionary of the provided info.

6. Unpacking Using `*` and `**`:

Question: You're given two lists and a dictionary:

```
nums1 = [1, 2, 3]  nums2 = [4, 5, 6]  data = {"x": 10, "y": 20}
```

Define a function called `unpacking_example` that takes in three positional arguments (`a`, `b`, `c`) and two keyword arguments (`x`, `y`). The function should return the sum of all its arguments. Use the given lists and dictionary to pass arguments to the function.

7. Basic Lambda Expression:

Question:

Write a lambda expression that takes two arguments and returns their sum.

8. Sorting with Lambda:

Question:

Given a list of tuples where each tuple contains a name and age, sort the list by age.(You can use `sorted()` function)

9. Filtering with Lambda:

Question:

Given a list of numbers, use a lambda expression to filter out the even numbers.(You can use `filter()` function)

10. Task: Define a function named operation that:

1. Accepts two mandatory positional arguments x and y.
2. Accepts an arbitrary number of additional numbers (using `*args`).
3. Accepts two keyword arguments action (with default value "add") and repeat (with default value 1).
4. Depending on the action value, performs the following:
 - "add": Add x, y, and all numbers in `*args`.
 - "multiply": Multiply x, y, and all numbers in `*args`.
5. The result of the above operation should be repeated repeat times in a list.

Note that you are free to use any built-in function required for this question