1. Create a function `greet` that takes two positional arguments: `name` and `greeting`, and returns a personalized greeting message.
2. Write a function `calculatePrice` that takes two arguments: `price` and `taxRate`. If `taxRate` is not provided, it should default to 5%.
3. Create a function `sumNumbers` that takes an unknown number of arguments using the rest operator and returns their sum.
4. Create a function `introduce` that takes three positional arguments: `firstName`, `lastName`, and `age`, and returns a sentence introducing the person.
5. Write a function `makeSandwich` that takes two arguments: `breadType` and `filling`. The default value for `breadType` should be "whole wheat".
6. Create a function `multiply` that takes two positional arguments `x` and `y`, with a default value for `y` as 1, and returns the result of multiplying them.
7. Write a function `getTotalPrice` that takes an unknown number of prices as arguments using the rest operator and returns the total price.
8. Create a function `buildSentence` that accepts three arguments: `subject`, `verb`, and `object`. If no `object` is passed, it should default to "something".
9. Write a function `addNumbers` that takes an unknown number of numbers using the rest operator and returns the sum.
10. Create a function `displayInfo` that takes three arguments: `name`, `age`, and `city`. If no city is provided, it should default to "Unknown".
11. Write a function `orderPizza` that takes two arguments: `size` and `topping`. If no `topping` is provided, it should default to "cheese".
12. Create a function `makeCoffee` that takes one required argument `type` and an unknown number of optional ingredients using the rest operator.
13. Write a function `introducePerson` that takes two positional arguments `name` and `profession`, with `profession` defaulting to "Student".
14. Create a function `findMax` that accepts an unknown number of numbers using the rest operator and returns the maximum value.
15. Write a function `buildPerson` that takes two arguments: `firstName` and `lastName`, and returns an object representing the person.
16. Create an object `car` with primitive properties: `brand`, `model`, and `year`.
17. Create an object `book` with non-primitive properties: `title`, `author`, and `reviews` (where `reviews` is an array of review objects).
18. Create an object `laptop` with both primitive (`brand`, `price`) and non-primitive (`features` - an array) properties.
19. Create an object `person` with a method `greet` that prints a greeting message using the `name` property.
20. Write a function `buildTeam` that takes an unknown number of team members using the rest operator and returns an array of their names.
21. Create a function `discountPrice` that takes two positional arguments: `price` and `discountPercentage`. If no discount is provided, it should default to 10%.

22. Write a function `describePet` that accepts two arguments: `petName` and `petType`. If no `petType` is provided, it should default to "dog".
23. Create a function `calculateArea` that accepts two arguments: `length` and `width`. If `width` is not provided, it should default to the same value as `length` (for a square).
24. Write a function `joinStrings` that accepts an unknown number of strings using the rest operator and returns them concatenated into one string.
25. Create a function `createUserProfile` that accepts three arguments: `name`, `email`, and `role`. If no `role` is provided, it should default to "user".
26. Write a function `divideNumbers` that takes two arguments: `a` and `b`, with `b` defaulting to 1 if not provided.
27. Create a function `combineArrays` that accepts an unknown number of arrays using the rest operator and returns one combined array.
28. Write a function `introduceEmployee` that takes two positional arguments: `employeeName` and `position`, with `position` defaulting to "intern".
29. Create a function `findAverage` that takes an unknown number of scores using the rest operator and returns the average score.
30. Write a function `buildCar` that accepts three arguments: `brand`, `model`, and `year`. If no `year` is provided, it should default to the current year.
31. Create a function `registerUser` that takes a required argument `username` and an unknown number of optional properties using the rest operator (e.g., email, age).
32. Write a function `favoriteFood` that takes one positional argument `food` and an unknown number of `ingredients` using the rest operator.
33. Create a function `assignTask` that accepts two arguments: `task` and `employee`, with `employee` defaulting to "Unassigned".
34. Write a function `createShoppingList` that accepts an unknown number of items using the rest operator and returns the shopping list as an array.
35. Create an object `movie` with primitive properties: `title`, `genre`, and `year`.
36. Create an object `restaurant` with non-primitive properties: `name`, `menu` (an array), and `ratings` (an array of rating objects).
37. Create an object `phone` with primitive properties `brand`, `model`, and `price`, and a method `getInfo` that returns a string with the phone's details.
38. Create an object `student` with both primitive (`name`, `age`) and non-primitive (`subjects` - an array) properties.
39. Write a function `greetFriends` that takes one required argument `greeting` and an unknown number of friend names using the rest operator.
40. Create an object `house` with primitive properties `address`, `size`, and `price`, and a method `getDetails` that returns a description of the house.