For Validating orders, the task can be decomposed into each of the validation requirements:

- 1: Pizzas in order must be from the same restaurant
- 2: Must not have more than 4 items in the order
- 3: Payment info must be valid
    - 3.1: CVV
        - It should have 3 or 4 digits.
        - It should have a digit between 0-9.
        - It should not have any alphabet or special characters.
    - 3.2: Card Number
    - 3.3: Expiry Date

Using these from the specification, we can see the individual tests we need in order to satisfy the overall requirement of validating orders.

The algorithm is constructed by creating these separate functions and calling them sequentially in an encapsulating method. Due to the modular approach, we can test each component rather easily as well as testing the final algorithm. We have decided that systemic testing is most suitable for this requirement as we can test the acceptable, extreme and exceptional inputs which will give good coverage over each of the components.

The first stage of testing is unit testing in which we do these individual component tests which verifies that the algorithm(s) are functional with respect to the specification. We then do the Unit test for the main algorithm as a whole which involves integrating the components into one controller method. We can perform this test by passing it a list of orders that are a mix of fully valid, not valid in all tests or not valid at all. We can then compare the output of valid orders to the orders we know to be valid to verify that the integration of the parts is working correctly.

Test Inputs

1:

- Order with one pizza
- Order with 4 pizzas same restaurant
- Order with 2 pizzas from different restaurants
- Order with 3 pizzas from one restaurant and 1 pizza from another

2:

- Order with no pizzas
- Order with 1 pizza
- Order with 4 pizzas
- Order with 5 pizzas

3.1:

- 3 digits, no special characters
- 4 digits, no special characters
- 2 digits, no special characters
- 5 digits, no special characters

- 3 digits, 3rd special character

3.2

- 16-digit length Luhn-valid
- 16-digit length Luhn-invalid
- 19-digit length Luhn-valid
- 19-digit length Luhn-invalid
- >19-digit length Luhn-valid
- >19-digit length Luhn-invalid
- <16-digit length Luhn-valid
- <16-digit length Luhn-invalid

3.3

- Expiry before order
- Expiry month of order
- Expiry day of order
- Expiry after order