Matrix (2D Array) Traversal

For each of the following create a function that takes a dynamic 2D array (assorted sizes of matrix) and returns a 1D array containing the output (items) from the given traversal. The best approach would be to create separate files for each of the numbered challenges and import these into a central main.py file for testing.

Sample matrix...

```
[['🍌','🍎','😀','🐉'],
 ['👺','🍺','🍩','🚴'],
 ['🚗','🐙','👤','🏝'],
 ['🌆','🖌','🕺','🍕']]
```

Challenges

1. Row-wise Traversal:
   a. Forward: Iterate through each row from left to right.
      ['🍌','🍎','😀','🐉','👺','🍺','🍩','🚴','🚗','🐙','👤','🏝','🌆','🖌','🕺','🍕']
   b. Reverse: Iterate through each row from right to left.
      ['🐉','😀','🍎','🍌','🚴','🍩','🍺','👺','🏝','👤','🐙','🚗','🍕','🕺','🖌','🌆']

2. Column-wise Traversal:
   a. Forward: Iterate through each column from top-left to bottom-right.
      ['🍌','👺','🚗','🌆','🍎','🍺','🐙','🖌','😀','🍩','👤','🕺','🐉','🚴','🏝','🍕']
   b. Reverse: Iterate through each column from bottom-left to top-right.
      ['🌆','🚗','👺','🍌','🖌','🐙','🍺','🍎','🕺','👤','🍩','😀','🍕','🏝','🚴','🐉']

3. Diagonal Traversal:
   a. Primary:
      i. Forward: Top-left to bottom-right.
         ['🍌','🍺','👤','🍕']
      ii. Reverse: Bottom-right to top-left.
         ['🍕','👤','🍺','🍌']
   b. Secondary:
      i. Forward: Top-right to bottom-left.
         ['🐉','🍩','🐙','🌆']
      ii. Reverse: Bottom-left to top-right.
         ['🌆','🐙','🍩','🐉']

4. Spiral Traversal:
   a. Forward: Move in a clockwise spiral pattern from the top-left.
      ['🍌','🍎','😀','🐉','🚴','🏝','🍕','🕺','🖌','🌆','🚗','👺','🍺','🍩','👤','🐙']
   b. Reverse: Move in an anti-clockwise spiral pattern from the top-left.
      ['🍌','👺','🚗','🌆','🖌','🕺','🍕','🏝','🚴','🐉','😀','🍎','🍺','🐙','👤','🍩']

5. Zigzag Traversal:
   a. Row First: Traverse rows left to right, then right to left, and repeat.
      ['🍌','👺','🍎','😀','🍺','🚗','🌆','🐙','🍩','🐉','🚴','👤','🖌','🕺','🏝','🍕']
   b. Column First: Traverse rows right to left, then left to right, and repeat.
      ['🍌','🍎','👺','🚗','🍺','😀','🐉','🍩','🐙','🌆','🖌','👤','🚴','🏝','🕺','🍕']

6. Snake Traversal:
    a. Forward: Traverse rows left to right, then right to left, and move to the next row.
    ['🍌','🍎','😀','🐉','🚴','🍩','🍺','👺','🚗','🐙','👤','🏝️','🍕','🕺','🏏','🌆']
    b. Reverse: Traverse rows right to left, then left to right, and move to the next row.
    ['🐉','😀','🍎','🍌','👺','🍺','🍩','🚴','🏝️','👤','🐙','🚗','🌆','🏏','🕺','🍕']
    c. Down: Traverse columns top to bottom, then right to left, and move to the next row.
    ['🍌','👺','🚗','🌆','🏏','🐙','🍺','🍎','😀','🍩','👤','🕺','🍕','🏝️','🚴','🐉']
    d. Up: Traverse columns bottom to top, then top to bottom, and move to the next column.
    ['🌆','🚗','👺','🍌','🍎','🍺','🐙','🏏','🕺','👤','🍩','😀','🐉','🚴','🏝️','🍕']

7. Boundary Traversal:
    a. Top:
        i. Forward: Left to right.
        ['🍌','🍎','😀','🐉']
        ii. Reverse: Right to left.
        ['🐉','😀','🍎','🍌']
    b. Right:
        i. Forward: Top to bottom.
        ['🐉','🚴','🏝️','🍕']
        ii. Reverse: Bottom to top.
        ['🍕','🏝️','🚴','🐉']
    c. Bottom:
        i. Forward: Left to right.
        ['🌆','🏏','🕺','🍕']
        ii. Reverse: Right to left.
        ['🍕','🕺','🏏','🌆']
    d. Left:
        i. Forward: Top to bottom.
        ['🍌','👺','🚗','🌆']
        ii. Reverse: Bottom to top.
        ['🌆','🚗','👺','🍌']
    e. Full:
        i. Forward: From top-left corner clockwise.
        ['🍌','🍎','😀','🐉','🚴','🏝️','🍕','🕺','🏏','🌆','🚗','👺']
        ii. Reverse: From top-left corner anticlockwise.
        ['🍌','👺','🚗','🌆','🏏','🕺','🍕','🏝️','🚴','🐉','😀','🍎']