2. String

Example 1: Check Email Address Validity

String1

```python
def is_valid_email(address):
    username = address.split('@')[0]
    domain_name = address.split('@')[-1]
    country_code = domain_name.split('.')

    check1_at_symbol = address.count('@') == 1
    check2_length = len(address) >= 8 and len(address) <= 30
    check3_first_letter = (address[0]).isalpha()
    check4_lower_complexity = False
    check4_upper_complexity = False
    check5_server_validity = '.' in domain_name and len(country_code[-1]) >= 2
    check6_country_code_validity = True

    for letter in username:
        if letter.islower():
            check4_lower_complexity = True
        elif letter.isupper():
            check4_upper_complexity = True
    for letter in country_code[-1][-2:]:
        if not letter.isalpha():
            check6_country_code_validity = False

    is_valid = (check1_at_symbol and check2_length and check3_first_letter and check4_lower_complexity \
                and check4_upper_complexity and check5_server_validity and check6_country_code_validity)
    return is_valid
```

Example 2: Capitalize Words

String2

```python
def capitalize_words(input_string):
    str_as_list = input_string.split(' ')
    str_as_list = [word for word in str_as_list if word != '']
    str_as_list = [word.capitalize() for word in str_as_list if word != '']
    # str_as_list = [word.capitalize() for word in input_string.split(' ') if word != '']

    res_str = ' '.join(str_as_list)
    return res_str
```

3. Lists

Example 1: Rotate Matrix

כתבו הפונקציה המקבלת מטריצה ומסובבת אותה ב- 90 מעלות עם כיוון השעון (ימינה). אין להשתמש ברשימות עזר. יש לבצע הכל על המטריצה המקורית וללא slicing.

Lists1

```python
def rotate_matrix_90_degrees_clockwise_v1(matrix):
    for i in range(len(matrix) // 2):
        for j in range(len(matrix) // 2):
            top_left = matrix[i][j]
            top_right = matrix[j][-i - 1]
            bottom_right = matrix[-i - 1][-j - 1]
            bottom_left = matrix[-j - 1][i]

            temp = top_left

            matrix[i][j] = bottom_left
            matrix[-j - 1][i] = bottom_right
            matrix[-i - 1][-j - 1] = top_right
            matrix[j][-i - 1] = temp

    return view_as_matrix(matrix)
```