

```
import sys
import ttt as program_file
```

```
#####
#####
# TODO:
# 1) PUT THIS FILE IN THE SAME FOLDER AS THE PYTHON FILE CONTAINING
YOUR SOLUTION #
# 1b) IF YOUR FILE_NAME IS DIFFERENT FROM ttt.PY THEN CHANGE ttt.PY
IN THE SECOND IMPORT STATEMENT TO THE NAME OF YOU FILE #
# 2) RUN THIS FILE
#
#####
#####@
```

```
# help functie voor het uitprinten van het resultaat van een test
met gegeven nummer en boolean die het succes van de test weergeeft
```

```
def print_result(test_nb, succes_boolean):
    test_string = 'Test ' + str(test_nb) + ':'
    if succes_boolean:
        print(test_string, 'correct')
    else:
        print(test_string, 'foutief')
```

```
# Testen voor de functie eentonige_lijst(l)
```

```
def test_eentonige_lijst():
    # Test 1
    expected = True
    succes = False
    try:
        succes = program_file.eentonige_lijst([]) == expected
    except Exception as e:
        print('Je programma goode volgende error:', e)
    print_result(1, succes)
    # Test 2
    expected = True
    try:
        succes = program_file.eentonige_lijst([1]) == expected
    except Exception as e:
        print('Je programma goode volgende error:', e)
    print_result(2, succes)
    # Test 3
    expected = True
    try:
        succes = program_file.eentonige_lijst([1, 1]) == expected
    except Exception as e:
        print('Je programma goode volgende error:', e)
    print_result(3, succes)
    # Test 4
    expected = False
    try:
```

```

        succes = program_file.eentonige_lijst([1, 2]) == expected
except Exception as e:
    print('Je programma gooide volgende error:', e)
print_result(4, succes)
# Test 5
expected = False
try:
    succes = program_file.eentonige_lijst([1, '1']) == expected
except Exception as e:
    print('Je programma gooide volgende error:', e)
print_result(5, succes)

# Testen voor de functie is_lege_matrix(m)
def test_is_lege_matrix():
    # Test 1
    expected = True
    succes = False
    try:
        succes = program_file.is_lege_matrix([]) == expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(1, succes)
    # Test 2
    expected = True
    try:
        succes = program_file.is_lege_matrix([[]]) == expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(2, succes)
    # Test 3
    expected = False
    try:
        succes = program_file.is_lege_matrix([1, 1]) == expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(3, succes)
    # Test 4
    expected = False
    try:
        succes = program_file.is_lege_matrix([1, 2]) == expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(4, succes)

# Testen voor de functie is_matrix(m)
def test_is_matrix():
    # Test 1
    expected = True
    succes = False
    try:
        succes = program_file.is_matrix([]) == expected
    except Exception as e:

```

```

    print('Je programma gooide volgende error:', e)
print_result(1, succes)
# Test 2
expected = True
try:
    succes = program_file.is_matrix([[[]]]) == expected
except Exception as e:
    print('Je programma gooide volgende error:', e)
print_result(2, succes)
# Test 3
expected = True
try:
    succes = program_file.is_matrix([[1]]) == expected
except Exception as e:
    print('Je programma gooide volgende error:', e)
print_result(3, succes)
# Test 4
expected = True
try:
    succes = program_file.is_matrix([[1, 'a']]) == expected
except Exception as e:
    print('Je programma gooide volgende error:', e)
print_result(4, succes)
# Test 5
expected = True
try:
    succes = program_file.is_matrix([[1, 'a'], [True, 1.2]]) ==
expected
except Exception as e:
    print('Je programma gooide volgende error:', e)
print_result(5, succes)
# Test 6
expected = False
try:
    succes = program_file.is_matrix([[1, 'a'], [True, 1.2],
["hello"]]) == expected
except Exception as e:
    print('Je programma gooide volgende error:', e)
print_result(6, succes)

# Testen voor de functie gelijke_rand_matrix(m)
def test_gelijke_rand_matrix():
    # Test 1
    expected = False
    succes = False
    try:
        succes = program_file.gelijke_rand_matrix([[1, 'a'], [True,
1.2]]) == expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(1, succes)
    # Test 2
    expected = False

```

```

    try:
        succes = program_file.gelijke_rand_matrix([[1, 1, 1, 1],
1],
                                                    [1, 1.2, True,
                                                    [1, 1, 2, 1]]) ==
expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(2, succes)
    # Test 3
    expected = True
    try:
        succes = program_file.gelijke_rand_matrix([[1, 1, 1, 1],
1],
                                                    [1, 1.2, True,
                                                    [1, 1, 1, 1]]) ==
expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(3, succes)
    # Test 4
    expected = True
    try:
        succes = program_file.gelijke_rand_matrix([[ 'a', 'a', 'a',
'a'],
                                                    [ 'a', 1.2, True,
'a'],
                                                    [ 'a', 1, 1, 'a'],
'a']]) == expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(4, succes)
    # Test 5
    expected = False
    try:
        succes = program_file.gelijke_rand_matrix([[ 'a', 'a', 'a',
'a'],
                                                    [ 'a', 1.2, True,
'a'],
                                                    [ 'a', 1, 1, 'a'],
'a']]) == expected
    except Exception as e:
        print('Je programma gooide volgende error:', e)
    print_result(5, succes)

# Testen voor de functie strip_matrix(m)
def test_strip_matrix():
    # Test 1
    expected = [ [1.2, True], [1, 1] ]
    succes = False
    try:

```

```

        succes = program_file.strip_matrix([[ 'a', 'a', 'a', 'a'],
                                             ['a', 1.2, True, 'a'],
                                             ['a', 1, 1, 'a'],
                                             ['a', 'b', 'a', 'a']])
== expected
    except Exception as e:
        print('Je programma goode volgende error:', e)
    print_result(1, succes)
    # Test 2
    expected = [ [1.2, True] ]
    try:
        succes = program_file.strip_matrix([[ 'a', 'a', 'a', 'a'],
                                             ['a', 1.2, True, 'a'],
                                             ['a', 'b', 'a', 'a']])
== expected
    except Exception as e:
        print('Je programma goode volgende error:', e)
    print_result(2, succes)
    # Test 3
    expected = False
    try:
        succes = program_file.strip_matrix([[1, 1, 1, 1]]) == [] or
program_file.strip_matrix([[1, 1, 1, 1]]) == [[]]
    except Exception as e:
        print('Je programma goode volgende error:', e)
    print_result(3, succes)

# Main part of the program: load all test cases and test them on
given solution
def run_all_tests():
    print('###')
    print('Testen voor de functie:', 'eentonige_lijst(l)')
    test_eentonige_lijst()
    print('###')
    print('Testen voor de functie:', 'is_lege_matrix(m)')
    test_is_lege_matrix()
    print('###')
    print('Testen voor de functie:', 'is_matrix(m)')
    test_is_matrix()
    print('###')
    print('Testen voor de functie:', 'gelijke_rand_matrix(m)')
    test_gelijke_rand_matrix()
    print('###')
    print('Testen voor de functie:', 'strip_matrix(m)')
    test_strip_matrix()
    print('###')

if __name__ == "__main__":
    run_all_tests()

```