

'''

Tussentijdse toets 2019-2020 en 2020-2021

Doel: kunnen opbouwen van (tot nu toe geleerde) data-structuren en kunnen werken met deze data-structuren.

'''

```
# Deze functie gaat na of alle elementen van een lijst gelijk zijn aan elkaar
# @param l - de lijst waarvan de elementen nagekeken worden
#         - de functie mag ervan uitgaan dat l verwijst naar een geldige (lege of niet-lege) lijst
# @return - True als alle elementen van l inderdaad gelijk zijn, anders False
#         - bemerk: de return-value van een lege lijst is True
def eentonige_list(l):
    pass

# Deze functie gaat na of de 2-dimensionale data-structuur leeg is of niet
# @param m - de 2-dimensionale data-structuur
#         - de functie mag ervan uitgaan dat m verwijst naar een geldige (lege of niet-lege) 2-dim
data-structuur
# @return - True als m een lege lijst is (dus: element bevat dat op zich een lege lijst is dus: [[]]);
anders False
def is_lege_matrix(m):
    pass

# Deze functie gaat na of de 2-dimensionale data-structuur een matrix is of niet
# @param m - de 2-dimensionale data-structuur
#         - de functie mag ervan uitgaan dat m verwijst naar een geldige (lege of niet-lege) 2-dim
data-structuur
# @return - True als alle 'rijen' van de data-structuur evenveel elementen bevatten, anders False
#         - bemerk: de return-value van een lege data-structuur is True
def is_matrix(m):
    pass

# Deze functie gaat na of alle elementen op de rand van een matrix gelijk zijn
# @param m - de matrix
#         - de functie mag ervan uitgaan dat m verwijst naar een (lege of niet-lege) matrix, zoals
hierboven gedefinieerd
# @return - True als alle elementen op de rand van m gelijk zijn; anders False
#         - bemerk: de return-value van een lege data-structuur is True
def gelijke_rand_matrix(m):
    pass

# Deze functie maakt een nieuwe matrix met alle elementen van een oorspronkelijke matrix zonder
de rand
# @param m - de oorspronkelijke matrix
#         - de functie mag ervan uitgaan dat m verwijst naar een geldige, niet-lege matrix, zoals
hierboven gedefinieerd
# @return - een matrix die alle elementen van m bevat zonder de rand-elementen
#         - bemerk: zie evt. de voorbeelden in de main-functie
def strip_matrix(m):
    pass
```

```

def main ():
    print("-- is_lege_matrix --")
    print(is_lege_matrix( [] ))      # True
    print(is_lege_matrix( [ [] ] ))  # True
    print(is_lege_matrix( [ [1, 1] ] )) # False
    print(is_lege_matrix( [ [1, 2] ] )) # False

    print("-- eentonige_lijt --")

    print(eentonige_lijt( [] ))      # True
    print(eentonige_lijt( [1] ))      # True
    print(eentonige_lijt( [1, 1] ))    # True
    print(eentonige_lijt( [1, 2] ))    # False
    print(eentonige_lijt( [1, '1'] ))  # False

    print("-- is_matrix --")

    print(is_matrix( [] ))            # True
    print(is_matrix( [ [] ] ))        # True
    print(is_matrix( [ [1] ] ))       # True
    print(is_matrix( [ [1, 'a'] ] ))   # True
    print(is_matrix( [ [1, 'a'], [True, 1.2] ] )) # True
    print(is_matrix( [ [1, 'a'], [True, 1.2], ["hello"] ] )) # False

    print("-- gelijke_rand_matrix --")

    print(gelijke_rand_matrix([[1, 'a'], [True, 1.2]])) # False

    print(gelijke_rand_matrix([[1, 1, 1, 1],
                                [1, 1.2, True, 1],
                                [1, 1, 2, 1]]))          # False

    print(gelijke_rand_matrix([[1, 1, 1, 1],
                                [1, 1.2, True, 1],
                                [1, 1, 1, 1]]))          # True

    print(gelijke_rand_matrix([[ 'a', 'a', 'a', 'a',
                                   'a', 1.2, True, 'a',
                                   'a', 1, 1, 'a',
                                   'a', 'a', 'a', 'a' ]])) # True

    print(gelijke_rand_matrix([[ 'a', 'a', 'a', 'a',
                                   'a', 1.2, True, 'a',
                                   'a', 1, 1, 'a',
                                   'a', 'b', 'a', 'a' ]])) # False

    print("-- strip_matrix --")

    print(strip_matrix([[ 'a', 'a', 'a', 'a',
                           'a', 1.2, True, 'a',
                           'a', 1, 1, 'a',
                           'a', 'b', 'a', 'a' ]]))          # [ [1.2, True], [1, 1] ]
    print(strip_matrix([[ 'a', 'a', 'a', 'a',
                           'a', 1.2, True, 'a' ],
                           [ 'a', 'a', 'a', 'a',
                           'a', 1.2, True, 'a' ],
                           [ 'a', 'a', 'a', 'a',
                           'a', 1.2, True, 'a' ],
                           [ 'a', 'a', 'a', 'a',
                           'a', 1.2, True, 'a' ] ]))

```

```
        ['a', 'b', 'a', 'a']))      # [ [1.2, True] ]
print(strip_matrix([[1, 1, 1, 1]]))  # een lege matrix, [] of [[]]
```

```
# VERANDER NIETS AAN DE LIJNEN HIERONDER, EN PLAATS ENKEL OPDRACHTEN
BINNEN DE FUNCTIES HIERBOVEN
if __name__ == "__main__":
    main()
```