## Exercise (H5.1)

Write a function that given a positive integer n, creates an nxn matrix where the value of A[i][j] is (i+1)*(j+1)

E.g., given n = 4, the matrix is:

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| 2 | 4 | 6 | 8 |
| 3 | 6 | 9 | 12 |
| 4 | 8 | 12 | 16 |

## Exercise (H5.2)

Write a function that sums two given matrices. Check first the compatibility of the dimensions for this operation, if not return None. Implement two versions:

1. by returning a new matrix, e.g., C = sum_matrix(A,B) → print(C)

2. inplace modification in the first matrix, e.g., sum_matrix(A,B) → print(A)

## Exercise (H5.3)

Write a function that maps two given lists into a dictionary where the first list elements serve as the keys and the second as the values . Check first the compatibility of the  dimensions for this operation, if not return None.

E.g., given [1, 2, 3, 4, 5] and [“a”, “b”, “c”, “d”, “e”], the function returns the dictionary {1: “a”, 2: “b”, 3: “c”, 4: “d”, 5: “e”}

## Exercise (H5.4)

Write a function that given two dictionaries, inserts the elements the second in the first. If an element is present in both, the function keeps the maximum value. In the dictionaries, the key is a string and the value is a number.

E.g., given {“a”: 5, “b”: 9, “c”: 20} and {“a”: 4, “b”: 11, “d”: 7}, the function returns the dictionary {“a”: 5, “b”: 11, “c”: 20, “d”: 7}

Implement two versions:

1. by returning a new dictionary, e.g., D_new = max_dict(D1, D2) → print(D_new)
2. inplace modification in the first dictionary, e.g., max_dict(D1, D2) → print(D1)

## Exercise (H5.5)

Assume to have three dictionaries which define a small database that stores the salaries of the employees working in one or more companies.

- employee → key: tuple(fiscal_code, ); value: tuple(name, surname, address, city, age)
- company → key: tuple(VAT_code, ); value: tuple(name, city)
- job → key: tuple(fiscal_code, VAT_code); value: tuple(salary)

'job' contains the associations between employees and companies, with the respective salary. Fill the dictionaries with some data, assuming that all tuple values are string.

Write functions that:

- creates a new dictionary where the key is the 'VAT_code' and the value is the sum of the salaries of all the employees working in that company
- given a fiscal_code of an employee, returns their total salary
- given a city name, returns the list of the employees working in that city

## Exercises (5.6 to 5.12)

- Repeat exercises h5.1 to h5.4 using list and dictionary comprehensions.
- Repeat exercises h4.3 and h4.5 using a dictionary and comprehension where appropriate.
- Write a function that given a list L and two integers a and b, returns a list whose elements are those of L appearing between a and b times in L.

  E.g., given L = [3,6,2,3,4,6,1,2,3,2,1,5,6], a = 2 and b = 3, the function returns the list
  [3, 6, 2, 1]