CS Homework #11

Deadline: December 14, 9:00 pm. Save your code as lastname_homework11.py and submit on Edmodo. Please, run your code before submitting. If you get an error, try to fix it before submitting your homework. If you get help from anyone, please, make sure that you actually understand the solution.

Task 1

Create an empty list A. Use for loop and range to append to A all even numbers from 30 to 10 (included), for example, 30, 28, 26, and so on. Create an empty list B. Append to B all even numbers from 40 to 20 (included).

Task 2

Create an empty list C. Write code that finds the numbers present in both A and B and append those numbers to C. Then create an empty list D. Write code that finds all *unique* numbers in A and B and append those numbers to D (that is, there should be no duplicates). Do NOT use Python sets to complete this task.

Task 3

Create a set setA which contains all elements of A. Create a set setB which contains all elements of B. Create a setC that contains common elements in A and B, or "intersection". For this task, do NOT use the previously created list C. Instead, use Python set intersection method which has the following format: x.intersection(y) -- finds intersection of x and y. Check if C and setC have the same elements (in no particular order).

Task 4

Create a setD that contains all elements in A and B excluding duplicates. For this task, use Python set union method which has the following format: x.union(y) -- finds union of x and y. Write code that verifies if D and setD have the same elements (again, the order does not matter).

Task 5

Can you solve *Task 4* without using x.update(y) method. Explore what this method does using setA and setB. What is the difference between x.union(y) and x.update(y). Try to figure this out without looking for an answer online. How can you use x.update(y) method to create setD that contains all elements of setA and setB, while preserving the original sets.

Task 6

Create a tuple of numbers from 1 to 8 and a tuple that contains the letters from “a” to “h”. Create a tuple that contains all coordinates of the chess board: ('a1', 'b1', 'c1', 'd1', 'e1', 'f1', 'g1', 'h1', 'a2', 'b2', 'c2', 'd2', 'e2', 'f2', 'g2', 'h2', 'a3', 'b3', 'c3', 'd3', 'e3', 'f3', 'g3', 'h3', 'a4', 'b4', 'c4', 'd4', 'e4', 'f4', 'g4', 'h4', 'a5', 'b5', 'c5', 'd5', 'e5', 'f5', 'g5', 'h5', 'a6', 'b6', 'c6', 'd6', 'e6', 'f6', 'g6', 'h6', 'a7', 'b7', 'c7', 'd7', 'e7', 'f7', 'g7', 'h7', 'a8', 'b8', 'c8', 'd8', 'e8', 'f8', 'g8', 'h8').

***Task 7**

Using the same original tuples of numbers from 1 to 8 and letters from “a” to “h”, create a nested (two dimensional) tuple that contains all coordinates of the chess board: (('a1', 'b1', 'c1', 'd1', 'e1', 'f1', 'g1', 'h1'), ('a2', 'b2', 'c2', 'd2', 'e2', 'f2', 'g2', 'h2'), ('a3', 'b3', 'c3', 'd3', 'e3', 'f3', 'g3', 'h3'), ('a4', 'b4', 'c4', 'd4', 'e4', 'f4', 'g4', 'h4'), ('a5', 'b5', 'c5', 'd5', 'e5', 'f5', 'g5', 'h5'), ('a6', 'b6', 'c6', 'd6', 'e6', 'f6', 'g6', 'h6'), ('a7', 'b7', 'c7', 'd7', 'e7', 'f7', 'g7', 'h7'), ('a8', 'b8', 'c8', 'd8', 'e8', 'f8', 'g8', 'h8')).
Verify yourtuple[2][2] == “c3” is True.