Задача 1

Проработать встроенние функции множеств. Встроенние функции можно взять на сайте (<https://www.programiz.com/python-programming/set>). Но на єтом сайте приведени примери

для списков, Задача переделать примери для множеств.

Задача 2

https://stackoverflow.com/questions/2541752/best-way-to-find-the-intersection-of-multiple-sets/2541814

Дано три множества

set1 = {1, 2, 3, 4}

set2 = {2, 3, 5, 6}

set3 = {3, 4, 6, 7}

Одним действием (одной строкой) виполнить intersection єтих множеств

Задача 3

https://www.journaldev.com/24683/python-set-difference

Дано три множества

set1 = {1, 2, 3, 4}

set2 = {2, 3, 5, 6}

set3 = {3, 4, 6, 7}

Одним действием (одной строкой) виполнить difference єтих множеств

Задача 4

Дано три множества

set1 = {1, 2, 3, 4}

set2 = {2, 3, 5, 6}

set3 = {3, 4, 6, 7}

Одним действием (одной строкой) виполнить union єтих множеств

Задача 5

https://pynative.com/python-set-exercise-with-solutions/

Добавить список элементов к заданному набору

sampleSet = {"Yellow", "Orange", "Black"}

sampleList = ["Blue", "Green", "Red"]

Задача 6

Вернуть новый набор идентичных предметов из заданных двух наборов

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

Задача 7

Возвращает новый набор со всеми элементами из обоих наборов, удаляя дубликаты.

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

Задача 8

Учитывая два набора Python, обновите первый набор элементами, которые существуют только в первом наборе, но не во втором наборе.

set1 = {10, 20, 30}

set2 = {20, 40, 50}

Задача 9

Удалите єлементи 10, 20, 30 из следующего набора

set1 = {10, 20, 30, 40, 50}

Задача 11

Проверьте, есть ли в двух наборах какие-либо общие элементы. Если да, отобразите общие элементы.

set1 = {10, 20, 30, 40, 50}

set2 = {60, 70, 80, 90, 10}

Задача 12

Обновите набор 1, добавив элементы из набора 2

set1 = {10, 20, 30, 40, 50}

set2 = {60, 70, 80, 90, 10}

Задача 13

Удалите элементы из set1, которые не являются общими для set1 и set2

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

Задача 14

Используя Все полученние знания по всем типам данних виполнить рефакторинг кода задачи с сложним списком из лекции 4. Код уменьшить до минимального количества строк

Задача 1

Проработать встроенние функции множеств. Встроенние функции можно взять на сайте (<https://www.programiz.com/python-programming/set>). Но на єтом сайте приведени примери

для списков, Задача переделать примери для множеств.

Задача 2

Дано три множества

set1 = {1, 2, 3, 4}

set2 = {2, 3, 5, 6}

set3 = {3, 4, 6, 7}

Одним действием (одной строкой) виполнить intersection єтих множеств

Задача 3

Дано три множества

set1 = {1, 2, 3, 4}

set2 = {2, 3, 5, 6}

set3 = {3, 4, 6, 7}

Одним действием (одной строкой) виполнить difference єтих множеств

Задача 4

Дано три множества

set1 = {1, 2, 3, 4}

set2 = {2, 3, 5, 6}

set3 = {3, 4, 6, 7}

Одним действием (одной строкой) виполнить union єтих множеств

Задача 5

Добавить список элементов к заданному набору

sampleSet = {"Yellow", "Orange", "Black"}

sampleList = ["Blue", "Green", "Red"]

Задача 6

Вернуть новый набор идентичных предметов из заданных двух наборов

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

Задача 7

Возвращает новый набор со всеми элементами из обоих наборов, удаляя дубликаты.

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

Задача 8

Учитывая два набора Python, обновите первый набор элементами, которые существуют только в первом наборе, но не во втором наборе.

set1 = {10, 20, 30}

set2 = {20, 40, 50}

Задача 9

Удалите єлементи 10, 20, 30 из следующего набора

set1 = {10, 20, 30, 40, 50}

Задача 11

Проверьте, есть ли в двух наборах какие-либо общие элементы. Если да, отобразите общие элементы.

set1 = {10, 20, 30, 40, 50}

set2 = {60, 70, 80, 90, 10}

Задача 12

Обновите набор 1, добавив элементы из набора 2

set1 = {10, 20, 30, 40, 50}

set2 = {60, 70, 80, 90, 10}

Задача 13

Удалите элементы из set1, которые не являются общими для set1 и set2

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

Задача 14

Используя Все полученние знания по всем типам данних виполнить рефакторинг кода задачи с сложним списком из лекции 4. Код уменьшить до минимального количества строк