**Словари и множества**

**Задание № 1**

Объедините словари (придумайте минимум 2 разных способа):

dict\_1 = {1: 'a', 2: 'b'}

dict\_2 = {2: 'c', 4: 'd'}

*Результат выполнения:*

{1: 'a', 2: 'c', 4: 'd'}

**Задание № 2**

Дан ряд словарей с пересекающимися ключами (значения - положительные числа). Например:

dict\_1 = {1: 12, 2: 33, 3: 10, 4: 10, 5: 2, 6: 90}

dict\_2 = {1: 12, 3: 7, 4: 1, 5: 2, 7: 112}

dict\_3 = {2: 3, 3: 3, 4: 60, 6: 8, 7: 25, 8: 71}

dict\_4 = {3: 1, 4: 13, 5: 31, 9: 9, 10: 556}

Задача:

Написать две функции, которые возвращают новый словарь, обрабатывая информацию следующим образом:

1. Функция max\_dict ()

Создает новый словарь по правилу:

- Если ключ встречается в нескольких словарях, выбирается максимальное значение среди всех значений этого ключа.

- Если ключ встречается только в одном словаре, он переносится в новый словарь без изменений.

Пример:

Для ключа «4» значения в словарях: 10, 1, 60, 13 → выбирается 60.

Для ключа «10» значение только в dict\_4 → переносится 556.

2. Функция sum\_dict ()

Создает новый словарь по правилу:

- Если ключ встречается в нескольких словарях, его значения суммируются.

- Если ключ уникален, он переносится без изменений.

Пример:

Для ключа «3» значения: 10 + 7 + 3 + 1 = 21.

Для ключа «9» значение только в dict\_4 → переносится 9.

Уточнения:

- Все ключи — целые числа (в исходном примере это 1, 2, 3, ...).

- Все значения — положительные числа.

*Тесты:*

print(max\_dct(dict\_1, dict\_2))

print(sum\_dct(dict\_1, dict\_4, dict\_3))

print(max\_dct(dict\_1, dict\_2, dict\_3, dict\_4))

print(sum\_dct(dict\_1, dict\_2, dict\_3, dict\_4))

*Результат выполнения:*

{1: 12, 2: 33, 3: 10, 4: 10, 5: 2, 6: 90, 7: 112}

{1: 12, 2: 36, 3: 14, 4: 83, 5: 33, 6: 98, 9: 9, 10: 556, 7: 25, 8: 71}

{1: 12, 2: 33, 3: 10, 4: 60, 5: 31, 6: 90, 7: 112, 8: 71, 9: 9, 10: 556}

{1: 24, 2: 36, 3: 21, 4: 84, 5: 35, 6: 98, 7: 137, 8: 71, 9: 9, 10: 556}

**Задание № 3**

Условие:

Дан список натуральных чисел. Необходимо преобразовать его в множество по следующим правилам:

- если число встречается один раз, оно включается в множество в исходном виде.

- если число повторяется несколько раз, то для каждого повторения начиная со второго, оно преобразуется в строку, состоящую из дублированного числа (количество дублирований равно номеру повторения).

Формат преобразования:

- первое вхождение числа: число остается как есть.

- второе вхождение: число преобразуется в строку из двух повторений.

- третье вхождение: строка из трёх повторений и т. д.

Задача:

Реализовать функцию set\_gen(), которая принимает список чисел и возвращает множество, сформированное по указанным правилам.

Порядок элементов в множестве не важен (поскольку set в Python неупорядочен).

Примечание:

Все числа во входном списке — натуральные (целые положительные).

Если число повторяется k раз, в множестве будет k элементов: 1 число и k-1 строк с дублированием.

*Тесты:*

list\_1 = [1, 1, 3, 3, 1]

list\_2 = [5, 5, 5, 5, 5, 5, 5]

list\_3 = [2, 2, 1, 2, 2, 5, 6, 7, 1, 3, 2, 2]

print(set\_gen(list\_1))

print(set\_gen(list\_2))

print(set\_gen(list\_3))

*Результат выполнения:*

{1, 3, '111', '33', '11'}

{'5555555', 5, '55', '55555', '5555', '555555', '555'}

{1, 2, 3, 5, 6, 7, '22', '2222', '22222', '222', '11', '222222'}

**Задание № 4**

Задача:

Требуется написать функцию superset(), которая принимает два множества и анализирует их соотношение. В зависимости от результата функция выводит одно из следующих сообщений:

- «Супермножество не обнаружено» – если ни одно из множеств не является супермножеством для другого.

- «Объект {X} является чистым супермножеством» – если одно из множеств (X) полностью включает в себя другое, но при этом имеет уникальные элементы.

- «Множества равны» – если множества идентичны.

Формат вывода:

В сообщении {X} подставляется само множество, которое является супермножеством.

Примечания:

- чистое супермножество – это множество, которое содержит все элементы другого множества, но имеет хотя бы один дополнительный элемент.

- если оба множества являются супермножествами друг для друга (т.е. они равны), выводится соответствующее сообщение.

- если ни одно из множеств не включает в себя другое, выводится сообщение об отсутствии супермножества.

*Тесты:*

set\_1 = {1, 8, 3, 5}

set\_2 = {3, 5}

set\_3 = {5, 3, 8, 1}

set\_4 = {90, 100}

superset(set\_1, set\_2)

superset(set\_1, set\_3)

superset(set\_2, set\_3)

superset(set\_4, set\_2)

*Результат выполнения:*

Объект {8, 1, 3, 5} является чистым супермножеством

Множества равны

Объект {8, 1, 3, 5} является чистым супермножеством

Супермножество не обнаружено