**Задача Python**

Написать функцию, которая:

1) принимает на вход итерируемый объект (list, tuple, np.array и т.п.) и число N - кол-во частей, на которые нужно разделить исходный итерируемый объект;

2) возвращает на выходе список из N-частей исходного итерируемого объекта.

Ограничения на результат:

1) Типы элементов, которые хранятся внутри исходного итерируемого объекта, не должны изменяться в результирующем списке:

[[1,'a','b'], [2, 'c', 'd'], [3, 'e', 'f']], N=2 -> [[1, 'a', 'b'], [2, 'c', 'd']], [[3, 'e', 'f']] - хорошо

[[1,'a','b'], [2, 'c', 'd'], [3, 'e', 'f']], N=2 -> [['1', 'a', 'b'], ['2', 'c', 'd']], [['3', 'e', 'f']] - плохо

2) Элементы исходного итерируемого объекта должны быть максимально равномерно распределены между частями:

[1,2,3,4,5,6,7,8,9], N=4 -> [[1,2,3], [4,5], [6,7], [8,9]] - хорошо

[1,2,3,4,5,6,7,8,9], N=4 -> [[1], [2,3], [4,5,6], [7,8,9]] – плохо

[1,2,3,4,5,6,7,8,9,10], N=4 -> [[1,2,3], [4,5, 6], [7,8], [9, 10]] – хорошо

[1,2,3,4,5,6,7,8,9,10], N=4 -> [[1,2,3], [4,5, 6], [7,8,9], [10]] – плохо

3) Разбиение должно сохранять порядок следования элементов в исходном списке

[1,2,3,4,5,6,7,8,9], N=4 -> [[1,2,3], [4,5], [6,7], [8,9]] - хорошо

[1,2,3,4,5,6,7,8,9], N=4 -> [[1,4,8], [2,5], [3,6], [7,9]] – плохо

**Задача SQL**

В таблице **n\_daily\_rests** содержится информация по остаткам товаров в магазинах сети Спортмастер:

**ID\_OBJ** - уникальный идентификатор магазина;

**ART** - артикул товара;

**DATEBEGIN** - дата начала временного промежутка, в котором остаток товара в заданном магазине по данному артикулу был равен AMOUNT;

**DATEEND** - дата конца временного промежутка (не включительно), в котором остаток товара в заданном магазине по заданному артикулу был равен AMOUNT.

В таблице **art\_category** содержится информация по категории товара

**ART** - артикул товара;

**CATEGORY** - категория товара;

Напишите следующие SQL запросы на Oracle SQL:

1. для каждой даты с 01.01.2020 по 31.01.2020 (не в совокупности, а по каждой отдельной дате) необходимо вывести сумму остатков по всем товарам, которые находились в магазинах, имеющих идентификаторы 171, 172, 174.
2. для каждой даты с 07.01.2020 по 07.01.2021 (не в совокупности, а по каждой отдельной дате) вывести идентификатор магазина (магазинов), который являлся третьим по величине остатков всех товаров на эту дату. Топ должен определяться только из тех магазинов, в которых на дату были остатки товаров из категории "Обувь" и были остатки как минимум по двум артикулам товаров из категории "Футболки".

**n\_daily\_rests**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID\_OBJ** | **ART** | **DATEBEGIN** | **DATEEND** | **AMOUNT** |
| 49 | MGFT296KOQ | 25.05.2020 | 31.05.2020 | 3 |
| 49 | MGFT296KOQ | 31.05.2020 | 10.06.2020 | 0 |
| 49 | MGFT296KOQ | 10.06.2020 | 01.01.2049 | 9 |
| 10 | MGFT296KOQ | 23.01.2019 | 26.04.2019 | 7 |
| 10 | MGFT296KOQ | 26.04.2019 | 01.01.2020 | 1 |

**art\_category**

|  |  |
| --- | --- |
| **ART** | **CATEGORY** |
| MGFT296KOQ | Футболки |