

Рубежный контроль**Вариант 6**

6	Дом	Улица
---	-----	-------

Вариант Б:

- 1) «Производитель» и «Деталь» связаны соотношением один-ко-многим. Выведите список всех связанных деталей и производителей, отсортированных по деталям, сортировка по производителям произвольная.
- 2) «Производитель» и «Деталь» связаны соотношением один-ко-многим. Выведите список производителей с количеством деталей у каждого производителя, отсортированный по количеству деталей.
- 3) «Производитель» и «Деталь» связаны соотношением многие-ко-многим. Выведите список всех деталей, у которых id госта заканчивается на «1», и названия их производителей.

Код программы

```
from operator import itemgetter

class Home:
    def __init__(self, id, home_number):
        self.id = id
        self.home_number=home_number

class Street:
    def __init__(self, id, streetName , ppl_count,Home_id):
        self.id = id
        self.streetName = streetName
        self.ppl_count = ppl_count
        self.Home_id = Home_id

class HomeStreet:
    def __init__(self, Home_id, Street_id):
        self.Home_id = Home_id
        self.Street_id = Street_id

Homes = [
    Home(1, 1),
```

```

    Home(2, 25),
    Home(3, 30),
]

Streets = [
    Street(1, "Bataskiy Proezd", 43, 1),
    Street(2, "Lugovoi proezd", 21, 2),
    Street(3, "Maryinskiy bulvar", 61, 3),
    Street(4, "Rahovo", 38, 3),
    Street(5, "Leninskiy Prospekt", 47, 1),
    Street(6, "Ulica Lenina", 42, 1)
]

Home_to_Street = [
    HomeStreet(1, 1),
    HomeStreet(2, 2),
    HomeStreet(3, 3),
    HomeStreet(3, 4),
    HomeStreet(1, 5),
]

def first_task(Streets_List):
    res_1 = sorted(Streets_List, key=itemgetter(2))
    return res_1

def second_task(Streets_List):
    res_2 = []
    temp_dict = dict()
    for i in Streets_List:
        if i[2] in temp_dict:
            temp_dict[i[2]] += 1
        else:
            temp_dict[i[2]] = 1
    for i in temp_dict.keys():
        res_2.append((i, temp_dict[i]))

    res_2.sort(key=itemgetter(1), reverse=True)
    return res_2

def third_task(Streets_List, end_ch):
    res_3 = [(i[0], i[2]) for i in Streets_List if str(i[1]).endswith(end_ch)]
    return res_3

def main():
    one_to_many = [(Street.streetName, Street.ppl_count, Home.home_number)
                    for Home in Homes
                    for Street in Streets

```

```

        if Street.Home_id == Home.id]

    many_to_many_temp = [(Home.home_number, connection.Home_id,
connection.Street_id)
        for Home in Homes
        for connection in Home_to_Street
        if connection.Home_id == Home.id]

    many_to_many = [(Street.streetName, Street.ppl_count, Home_adress)
        for Home_adress, Home_id, Street_id in many_to_many_temp
        for Street in Streets if Street.id == Street_id]

    print('Задание Б1')
    print(first_task(one_to_many))

    print("\nЗадание Б2")
    print(second_task(one_to_many))

    print("\nЗадание Б3")
    print(third_task(many_to_many, '1'))

if __name__ == '__main__':
    main()

```

Результаты выполнения

```

C:\Users\demoh\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\demoh\PycharmProjects\pythonProject\rk1.py
Задание Б1
[('Bataskiy Proezd', 43, 1), ('Leninskiy Prospekt', 47, 1), ('Ulica Lenina', 42, 1), ('Lugovoi proezd', 21, 25), ('Maryinskiy bulvar', 61, 30), ('Rahovo', 38, 30)]

Задание Б2
[(1, 3), (30, 2), (25, 1)]

Задание Б3
[('Lugovoi proezd', 25), ('Maryinskiy bulvar', 30)]

```