

Задание (Вариант Б6):

1. Улица и Дом связаны соотношением один-ко-многим. Выведите список всех связанных улиц и домов, отсортированный по номерам домов, сортировка по улицам произвольная.
2. Улица и Дом связаны соотношением один-ко-многим. Выведите список улиц с количеством домов на каждой улице, отсортированный по количеству домов.
3. Улица и Дом связаны соотношением многие-ко-многим. Выведите список всех домов с нечетными номерами и названия их улиц.

Код:

```
#!/usr/bin/env python
```

```
class House:
```

```
    """Дом"""
```

```
    def __init__(self, id, number, street_id):
```

```
        self.id = id
```

```
        self.number = number
```

```
        self.street_id = street_id
```

```
class Street:
```

```
    """Улица"""
```

```
    def __init__(self, id, name):
```

```
        self.id = id
```

```
        self.name = name
```

```
class HouseStreet:
```

```
    """
```

```
    'Дома на улице' для реализации
```

```
    связи многие-ко-многим
```

```
    """
```

```
    def __init__(self, street_id, house_id):
```

```
        self.street_id = street_id
```

```
        self.house_id = house_id
```

```
streets = [
```

```
    Street(1, "Бабаевская улица"),
```

```
    Street(2, "Бабьегородский 1-й, переулок"),
```

```
    Street(3, "Бабьегородский 2-й, переулок"),
```

```
Street(11, "Банковский переулок"),
Street(22, "Банный переулок"),
Street(33, "Банный проезд"),
]
```

```
houses = [
    House(1, 8, 1),
    House(2, 9, 1),
    House(3, 10, 2),
    House(4, 11, 3),
    House(5, 12, 11),
    House(6, 13, 33),
    House(7, 14, 11),
]
```

```
house_streets = [
    HouseStreet(1, 1),
    HouseStreet(2, 2),
    HouseStreet(2, 3),
    HouseStreet(3, 4),
    HouseStreet(3, 5),

    HouseStreet(11, 1),
    HouseStreet(22, 2),
    HouseStreet(33, 3),
    HouseStreet(33, 4),
    HouseStreet(33, 5),
]
```

```
def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(h.number, s.name)

                    for s in streets
                    for h in houses
                    if h.street_id == s.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(s.name, hs.street_id, hs.house_id)

                          for s in streets
```

```

for hs in house_streets
if s.id == hs.street_id]

many_to_many = [(h.number, street_name)

for street_name, street_id, house_id in many_to_many_temp
for h in houses if h.id == house_id]

print("Задание Б1")
res_11 = sorted(one_to_many, key=lambda x: x[0])
print(res_11)

print()
print("Задание Б2")
res_12_unsorted = []
# Перебираем все улицы
for s in streets:
    # Список домов улицы
    houses_tmp = list(filter(lambda x: x[1] == s.name, one_to_many))
    # Если улица не пустая
    if len(houses_tmp) > 0:
        res_12_unsorted.append((s.name, len(houses_tmp)))

# Сортировка по количеству домов
res_12 = sorted(res_12_unsorted, key=lambda x: x[1], reverse=True)
print(res_12)

print()
print("Задание Б3")
res_13 = { }
# Перебираем все дома
for h in houses:
    if h.number % 2 != 0:
        # Список домов улицы
        houses_tmp = list(filter(lambda x: x[0] == h.number, many_to_many))
        # Только имена улиц
        street_names = [x for _, x in houses_tmp]
        # Добавляем результат в словарь
        # Ключ - номер дома, значение - список улиц
        res_13[h.number] = street_names

print(res_13)

```

```
if __name__ == '__main__':  
    main()
```

Результат работы:

Задание Б1

[(8, 'Бабаевская улица'), (9, 'Бабаевская улица'), (10, 'Бабыгородский 1-й, переулок'), (11, 'Бабыгородский 2-й, переулок'), (12, 'Банковский переулок'), (13, 'Банный проезд'), (14, 'Банковский переулок')]

Задание Б2

[('Бабаевская улица', 2), ('Банковский переулок', 2), ('Бабыгородский 1-й, переулок', 1), ('Бабыгородский 2-й, переулок', 1), ('Банный проезд', 1)]

Задание Б3

{9: ['Бабыгородский 1-й, переулок', 'Банный переулок'], 11: ['Бабыгородский 2-й, переулок', 'Банный проезд'], 13: []}