

```
# используется для сортировки
from operator import itemgetter
from itertools import groupby

class Doc:
    """Документ"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Sec:
    """Раздел"""
    def __init__(self, id, name, leng, doc):
        self.id = id
        self.name = name
        self.leng = leng
        self.doc = doc

class SecDoc:
    """
    'Сотрудники отдела' для реализации
    связи многие-ко-многим
    """
    def __init__(self, sec_id, doc_id):
        self.doc_id = doc_id
        self.sec_id = sec_id

# Документы
docs = [
    Doc(1, 'Договор от 21.10.21'),
    Doc(2, 'Соглашение от 29.09.21'),
    Doc(3, 'Устав от 11.05.19'),

    Doc(6, 'Договор от 13.07.22'),
    Doc(7, 'Соглашение от 31.03.22'),
    Doc(8, 'Устав от 01.01.99'),
]

# Разделы
secs = [
    Sec(1, 'Оглавление', 1, 1),
    Sec(2, 'Автоматизация', 10, 1),
    Sec(3, 'Стоимость', 2, 2),
    Sec(4, 'Ансамбль', 5, 2),
    Sec(5, 'Условия', 7, 2),
```

```

        Sec(6, 'Предметы и виды деятельности', 3, 3),
        Sec(7, 'Участники', 11, 3),
        Sec(8, 'Уставной Капитал', 1, 8),
        Sec(9, 'преамбула', 5, 7),
        Sec(10, 'реквизиты', 6, 1),
    ]

secs_docs = [
    SecDoc(1,1),
    SecDoc(2,1),
    SecDoc(3,2),
    SecDoc(4,2),
    SecDoc(5,2),
    SecDoc(6,3),
    SecDoc(7,3),
    SecDoc(8,8),
    SecDoc(9,7),
    SecDoc(10,1),

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

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

    # Соединение данных один-ко-многим
    one_to_many = [(e.name, e.leng, d.name)
                    for d in docs
                    for e in secs
                    if e.doc == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, ed.doc_id, ed.sec_id)
                           for d in docs
                           for ed in secs_docs
                           if d.id==ed.doc_id]

    many_to_many = [(e.name, e.leng, doc_name)
                     for doc_name, _, dep_id in many_to_many_temp
                     for e in secs if e.id==dep_id]

```

```

print('Задание B1')
print("\n".join(map(lambda x: f"Документ : {x[2]}, Раздел : {x[0]}, Длинной {x[1]}", filter(lambda x: x[0].lower()[0] == 'a', one_to_many))))

print("\nЗадание B2")
print("\n".join("Документ: {2}, Раздел минимальной длины: {0}, Длина {1}".format(*sorted(i[1], key= lambda x: x[1])[0]) for i in groupby(one_to_many, lambda x: x[2])))

print("\nЗадание B3")
print("\n".join(map(lambda x: f"Раздел: {x[0]}, длина: {x[1]}, Документ: {x[2]}", sorted(many_to_many))))

if __name__ == '__main__':
    main()

```

Результаты выполнения:

```

Задание B1
Документ : Договор от 21.10.21, Раздел : Автоматизация, Длинной 10
Документ : Соглашение от 29.09.21, Раздел : Ансамбль, Длинной 5

Задание B2
Документ: Договор от 21.10.21, Раздел минимальной длины: Оглавление, Длина 1
Документ: Соглашение от 29.09.21, Раздел минимальной длины: Стоимость, Длина 2
Документ: Устав от 11.05.19, Раздел минимальной длины: Предметы и виды деятельности, Длина 3
Документ: Соглашение от 31.03.22, Раздел минимальной длины: преамбула, Длина 5
Документ: Устав от 01.01.99, Раздел минимальной длины: Уставной Капитал, Длина 1

Задание B3
Раздел: Автоматизация, длина: 10, Документ: Договор от 21.10.21
Раздел: Ансамбль, длина: 5, Документ: Соглашение от 29.09.21
Раздел: Оглавление, длина: 1, Документ: Договор от 21.10.21
Раздел: Предметы и виды деятельности, длина: 3, Документ: Устав от 11.05.19
Раздел: Стоимость, длина: 2, Документ: Соглашение от 29.09.21
Раздел: Условия, длина: 7, Документ: Соглашение от 29.09.21
Раздел: Уставной Капитал, длина: 1, Документ: Устав от 01.01.99
Раздел: Участники, длина: 11, Документ: Устав от 11.05.19
Раздел: преамбула, длина: 5, Документ: Соглашение от 31.03.22
Раздел: реквизиты, длина: 6, Документ: Договор от 21.10.21

```