

```
from operator import itemgetter

class Section():
    def __init__(self, id, name, responsible, date, document_id):
        self.id = id
        self.name = name
        self.responsible = responsible #фамилия ответственного
        self.date = date
        self.document_id = document_id

class Document():
    def __init__(self, id, name):
        self.id = id
        self.name = name

class SD():
    def __init__(self, sec_id, doc_id):
        self.sec_id = sec_id
        self.doc_id = doc_id

Docs = [
    Document(1, 'Проект 1'),
    Document(2, 'Проект 2'),
    Document(3, 'Проект 3'),
    Document(4, 'Проект 4'),
]

Sections = [
    Section(1, 'ТЗ', 'Лобанов', '2020.10.23', 1),
    Section(2, 'ТЗ', 'Лобанов', '2020.10.25', 2),
    Section(3, 'ТЗ', 'Демирев', '2020.10.22', 3),
    Section(4, 'ТЗ', 'Демирев', '2020.10.28', 4),
    Section(5, 'ТО', 'Токаев', '2020.10.23', 1),
    Section(6, 'ТО', 'Токаев', '2020.10.23', 2),
    Section(7, 'ТО', 'Арзаманов', '2020.10.23', 3),
    Section(8, 'ТО', 'Арзаманов', '2020.10.23', 4),
    Section(9, 'РПЗ', 'Арзаманов', '2020.10.1', 4)
]

Sec_doc = [
    SD(1, 1),
    SD(1, 2),
    SD(3, 2),
    SD(3, 3),
    SD(5, 3),
    SD(5, 4),
    SD(7, 4),
    SD(7, 1)
]

def main():
```

```

# Соединение данных один-ко-многим
one_to_many = [(e.name, e.responsible, d.name)
                for d in Docs
                for e in Sections
                if e.document_id == d.id]

print()
print('Задание B1')
print(sorted(one_to_many, key=itemgetter(1)))

one_to_many_2 = set()

#Выводим проекты с количеством разделов

for i in Docs:
    proj = ''
    k = 0
    for j in Sections:
        if j.document_id == i.id:
            if proj == '':
                proj = i.name
                k += 1
            else:
                k += 1
                continue
    one_to_many_2.add((proj, k))

print()
print('Задание B2')
print(sorted(one_to_many_2, key=itemgetter(1)))

many_to_many = {}

for i in Sec_doc:
    length = len(Sections[i.sec_id-1].responsible)
    if Docs[i.doc_id-1].name in many_to_many.keys():
        if Sections[i.sec_id-1].responsible[length-
1] == 'B' and Sections[i.sec_id-1].responsible[length-2] == 'o':
            many_to_many[Docs[i.doc_id-1].name].add((Sections[i.sec_id-
1].responsible))
        else:
            if Sections[i.sec_id-1].responsible[length-
1] == 'B' and Sections[i.sec_id-1].responsible[length-2] == 'o':
                many_to_many[Docs[i.doc_id-1].name] = set()
                many_to_many[Docs[i.doc_id-1].name].add((Sections[i.sec_id-
1].responsible))

print()
print('Задание B3')
print(many_to_many)

```

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

Задание В1

```
[('ТО', 'Арзаманов', 'Проект 3'), ('ТО', 'Арзаманов', 'Проект 4'), ('РПЗ', 'Арзаманов', 'Проект 4'), ('ТЗ', 'Демиров', 'Проект 3'), ('ТЗ', 'Демиров', 'Проект 4'), ('ТЗ', 'Лобанов', 'Проект 1'), ('ТЗ', 'Лобанов', 'Проект 2'), ('ТО', 'Токаев', 'Проект 1'), ('ТО', 'Токаев', 'Проект 2')]
```

Задание В2

```
[('Проект 1', 2), ('Проект 2', 2), ('Проект 3', 2), ('Проект 4', 3)]
```

Задание В3

```
{'Проект 1': {'Арзаманов', 'Лобанов'}, 'Проект 2': {'Лобанов'}, 'Проект 4': {'Арзаманов'}}
```

PS C:\Users\danch>