Текст программы

```
# используется для сортировки
from operator import itemgetter

# -------------------------nop - number of pages

# Класс "книга"
class Book:
    def __init__(self, id, name, nop, store_id):
        self.id = id
        self.name = name
        # Колличество пропущенных учебных дней
        self.nop = nop
        self.store_id = store_id


# Класс "книжный магазин"
class City_store:
    def __init__(self, id, name):
        self.id = id
        self.name = name



class BookStore:
    def __init__(self, store_id, book_id):
        self.store_id = store_id
        self.book_id = book_id


# Книги
books = [
    Book(1, 'Оно', 1138, 1),
    Book(2, 'Преступление и наказание', 574, 1),
    Book(3, 'Гамлет', 300, 2),
    Book(4, 'Убийство в Восточном экспрессе', 317, 2),
    Book(5, 'Молчание ягнят', 496, 3),
]


# Магазины
storees = [
    City_store(1, 'Ходасевич'),
    City_store(2, 'Пушкинская лавка'),
    City_store(3, 'Гиперион'),

    City_store(11, 'Художественная литература'),
    City_store(22, 'Фаланстер'),
    City_store(33, 'Циолковский'),
]
```

```
books_storees = [
    BookStore(1,1),
    BookStore(1,2),
    BookStore(2,3),
    BookStore(2,4),
    BookStore(3,5),

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


def main():

    # Соединение данных один-ко-многим
    one_to_many = [(p.name, p.nop, c.name)
        for c in storees
        for p in books
        if p.store_id==c.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(c.name, pc.store_id, pc.book_id)
        for c in storees
        for pc in books_storees
        if c.id==pc.store_id]

    many_to_many = [(p.name, p.nop, store_name)
        for store_name, store_id,book_id in many_to_many_temp
        for p in books if p.id==book_id]



    print('Задание А1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)


    print('\nЗадание А2')
    res_12_unsorted = []
    # Перебираем все классы
    for c in storees:
        # Список книг магазина
        c_books = list(filter(lambda i: i[2]==c.name,
one_to_many))
        # Если магазин не пустой
        if len(c_books) > 0:
```

```
            # количество страниц книг магазина
            c_nops = [nop for _,nop,_ in c_books]
            # Суммарное количество страниц книг магазина
            c_nops_sum = sum(c_nops)
            res_12_unsorted.append((c.name, c_nops_sum))
    # Сортировка по суммарному количеству
    res_12 = sorted(res_12_unsorted, key=itemgetter(1),
reverse=True)
    print(res_12)


    print('\nЗадание А3')
    res_13 = {}
    # Перебираем все классы
    for c in storees:
        if 'X' in c.name:
            # Список книг магазина
            c_books = list(filter(lambda i: i[2]==c.name,
many_to_many))
            # Только название книг
            c_books_names = [x for x,_,_ in c_books]
            # Добавляем результат в словарь
            # ключ - класс, значение - список фамилий
            res_13[c.name] = c_books_names
    print(res_13)


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

Результаты выполнения:

```
Задание А1
[('Молчание ягнят', 496, 'Гиперион'), ('Гамлет', 300, 'Пушкинская
лавка'), ('Убийство в Восточном экспрессе', 317, 'Пушкинская
лавка'), ('Оно', 1138, 'Ходасевич'), ('Преступление и наказание',
574, 'Ходасевич')]

Задание А2
[('Ходасевич', 1712), ('Пушкинская лавка', 617), ('Гиперион',
496)]

Задание А3
{'Ходасевич': ['Оно', 'Преступление и наказание'], 'Художественная
литература': ['Оно', 'Преступление и наказание']}
```