```python
# используется для сортировки
from operator import itemgetter


class Chapter:
    """Глава"""
    def __init__(self, id, chap, page, book_id):
        self.id = id
        self.chap = chap
        self.page = page
        self.book_id = book_id

class Book:
    """Книга"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

# Книги
books = [
    Book(11, 'Дюна'),
    Book(22, 'Мессия дюны'),
    Book(33, 'Дети дюны'),
    Book(44, 'Бог-император дюны'),
    Book(55, 'Еретики дюны'),
]

# Главы
chapters = [
    Chapter(1, 'Глава 1', 6, 11),
    Chapter(2, 'Глава 2', 35, 22),
    Chapter(3, 'Глава 3', 94, 33),
    Chapter(4, 'Глава 4', 143, 44),
    Chapter(5, 'Глава 5', 254, 55),
]

def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one_to_many = [(c.chap, c.page, b.name)
        for b in books
        for c in chapters
        if c.book_id == b.id]

    print('Задание 1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание 2')
    res_12_unsorted = []
    # Перебираем все книги
    for b in books:
```

```
        c_chapters = list(filter(lambda i: i[2]==b.name, one_to_many))
        if len(c_chapters) > 0:
            # Страница главы книги
            c_pages = [page for _,page,_ in c_chapters]
            # Суммарная страница
            c_pages_sum = sum(c_pages)
            res_12_unsorted.append((b.name, c_pages_sum))


    # Сортировка по суммарной странице
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)


if __name__ == '__main__':
    main()


""" РЕЗУЛЬТАТ
Задание 1
[('Глава 4', 143, 'Бог-император дюны'), ('Глава 3', 94, 'Дети дюны'),
  ('Глава 1', 6, 'Дюна'), ('Глава 5', 254, 'Еретики дюны'), ('Глава 2', 35,
  'Мессия дюны')]

Задание 2
[('Еретики дюны', 254), ('Бог-император дюны', 143), ('Дети дюны', 94),
  ('Мессия дюны', 35), ('Дюна', 6)]

 """
```