Группа ИУ5-31Б, Шибанова Варвара, Вариант: 24А

```
# используется для сортировки
from operator import itemgetter

class Chapter:
    """Глава"""
    def __init__(self, id, tit, pag, b_id):
        self.id = id
        self.tit = tit
        self.pag = pag
        self.b_id = b_id

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

class ChapBook:
    """
    'Главы книги' для реализации
    связи многие-ко-многим
    """
    def __init__(self, b_id, chap_id):
        self.b_id = b_id
        self.chap_id = chap_id

# Книги
books = [
    Book(1, 'Маленький принц'),
    Book(2, 'Американский психопат'),
    Book(3, 'Палач'),

    Book(11, 'Гарри Поттер и узник Азкабана'),
    Book(22, 'Путешествие к центру Земли'),
    Book(33, 'Обломов'),
]

# Главы
chapters = [
    Chapter(1, 'Введение', 10, 3),
    Chapter(2, 'Заключение', 12, 1),
    Chapter(3, 'Глава 6', 20, 2),
    Chapter(4, 'Роза', 15, 1),
    Chapter(5, 'Расследование', 27, 3),
]

chap_b = [
    ChapBook(1,4),
```

```
    ChapBook(2,3),
    ChapBook(3,5),
    ChapBook(3,1),
    ChapBook(3,2),

    ChapBook(11,1),
    ChapBook(22,2),
    ChapBook(33,1),
    ChapBook(33,2),
    ChapBook(22,1),
]

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

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(b.name, c.b_id, c.chap_id)
        for b in books
        for c in chap_b
        if b.id==c.b_id]

    many_to_many = [(c.tit, c.pag, book_name)
        for book_name, b_id, chap_id in many_to_many_temp
        for c in chapters if c.id==chap_id]

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

# сортировка книг по суммарному количеству страниц
    print('\nЗадание А2')
    res_12_unsorted = []
    for b in books:
        b_chaps = list(filter(lambda i: i[2]==b.name, one_to_many))
        if len(b_chaps) > 0:
            b_pag = [pag for _,pag,_ in b_chaps]
            b_pag_sum = sum(b_pag)
            res_12_unsorted.append((b.name, b_pag_sum))

    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)

#список всех книг в названии которых присутствует "и"
    print('\nЗадание А3')
    res_13 = {}
```

```
    for b in books:
        if 'и' in b.name:
            b_chaps = list(filter(lambda i: i[2]==b.name, many_to_many))
            b_chaps_names = [x for x,_,_ in b_chaps]
            res_13[b.name] = b_chaps_names

    print(res_13)

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

```
Задание А1
[('Глава 6', 20, 'Американский психопат'), ('Заключение', 12, 'Маленький принц'), ('Роза', 1
5, 'Маленький принц'), ('Введение', 10, 'Палач'), ('Расследование', 27, 'Палач')]

Задание А2
[('Палач', 37), ('Маленький принц', 27), ('Американский психопат', 20)]

Задание А3
{'Маленький принц': ['Роза'], 'Американский психопат': ['Глава 6'], 'Гарри Поттер и узник Аз
кабана': ['Введение'], 'Путешествие к центру Земли': ['Заключение', 'Введение']}
PS C:\Users\VARVARA\Desktop\Парадигмы в конструкции языков програмирования\Labs.py>
```