```python
# используется для сортировки
#Conductor/Orchestra
from operator import itemgetter


class Conductor:
    """Дирижер"""
    def __init__(self, id, age, exp, orchestra_id):
        self.id = id
        self.age = age
        self.exp = exp
        self.orchestra_id = orchestra_id #какой Orchestra соответсвует


class Orchestra:
    """Оркестр"""
    def __init__(self, id, name):
        self.id = id
        self.name = name


class ConductorOrchestra:
    """
    Дирижеры разных оркестров для реализации
    связи многие-ко-многим
    """
    def __init__(self, orchestra_id, conductor_id):
        self.orchestra_id = orchestra_id
        self.conductor_id = conductor_id


# Оркестры
Orchestras = [
    Orchestra(1, 'Хихи-Хаха'),
    Orchestra(2, 'Хих-Хах'),
    Orchestra(3, 'Хи-Хик'),


    Orchestra(11, 'Хе-Хе'),
    Orchestra(22, 'Хо-Хо'),
    Orchestra(33, 'Ха-Ха'),
]


# Дирижеры
Conductors = [
    Conductor(1, 'Первый', 100, 37),
    Conductor(2, 'Второй', 52, 18),
    Conductor(3, 'Третий 6', 30, 22),
    Conductor(4, 'Четвертый', 45, 10),
    Conductor(5, 'Пятый', 71, 32),
]


Conductor_Orchestra = [
    ConductorOrchestra(1,4),
    ConductorOrchestra(2,3),
    ConductorOrchestra(3,5),
    ConductorOrchestra(3,1),
```

```
    ConductorOrchestra(3,2),


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


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


    # Соединение данных один-ко-многим
    one_to_many = [(c.age, c.exp, o.name)
        for o in Orchestras
        for c in Conductors
        if c.orchestra_id == o.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(o.name, c.orchestra_id, c.conductor_id)
        for o in Orchestras
        for c in Conductor_Orchestra
        if o.id==c.orchestra_id]

    many_to_many = [(c.age, c.exp, orchestra_name)
        for orchestra_name, orchestra_id, conductor_id in many_to_many_temp
        for c in Conductors if c.id == conductor_id]


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

    print('\nЗадание А2')
    res_12_unsorted = []
    for b in Orchestras:
        b_Conductors = list(filter(lambda i: i[2]==b.name, one_to_many))
        if len(b_Conductors) > 0:
            b_pag = [pag for _,pag,_ in b_Conductors]
            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)


    print('\nЗадание А3')
    res_13 = {}
    for b in Orchestras:
        if 'и' in b.name:
            b_Conductors = list(filter(lambda i: i[2]==b.name, many_to_many))
            b_Conductors_names = [x for x,_,_ in b_Conductors]
            res_13[b.name] = b_Conductors_names
```

```
    print(res_13)


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