

```

# используется для сортировки
from operator import itemgetter
from numpy import unique

class Group:
    """Сотрудник"""
    def __init__(self, id, elder, count, dep_id):
        self.id = id
        self.elder = elder
        self.count = count
        self.dep_id = dep_id

class Department:
    """Отдел"""

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

class GrDep:
    """
    'Сотрудники отдела' для реализации
    СВЯЗИ МНОГИЕ-КО-МНОГИМ
    """

    def __init__(self, dep_id, emp_id):
        self.dep_id = dep_id
        self.emp_id = emp_id

# Отделы
deps = [
    Department(1, 'ИУ5'),
    Department(2, 'ИУ6'),
    Department(3, 'РЛ6'),

    Department(11, 'ФН4'),
    Department(22, 'ИУ12'),
    Department(33, 'Э3'),
]

# Сотрудники
emps = [
    Group(1, 'Артамонов', 25, 1),
    Group(2, 'Белодедов', 35, 2),
    Group(3, 'Иваненко', 45, 3),
    Group(4, 'Иванов', 35, 3),
    Group(5, 'Иванин', 25, 3),
]

emps_deps = [
    GrDep(1, 1),
    GrDep(2, 2),
    GrDep(3, 3),
    GrDep(3, 4),
    GrDep(3, 5),

    GrDep(11, 1),
    GrDep(22, 2),
    GrDep(33, 3),
    GrDep(33, 4),

```

```

GrDep(33, 5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(e.elder, e.count, d.name)
                    for d in deps
                    for e in emps
                    if e.dep_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, ed.dep_id, ed.emp_id)
                           for d in deps
                           for ed in emps_deps
                           if d.id == ed.dep_id]

    many_to_many = [(e.elder, e.count, dep_name)
                     for dep_name, dep_id, emp_id in many_to_many_temp
                     for e in emps if e.id == emp_id]

    def A_1(one_to_many: list):
        return list(sorted(filter(lambda x: x[2][0] == 'И', one_to_many)))
    def A_2(one_to_many: list):
        return unique(sorted([[c, sum(j for i, j, k in list(filter(lambda x:
x[2] == c, one_to_many)))] for a, b, c in one_to_many]), axis=0)
    def A_3(many_to_many: list):
        return sorted(many_to_many, key=lambda x: x[2])

    print('Задание A1')
    [print(f'{caf} - {elder}, {count} студентов') for elder, count, caf in
A_1(one_to_many)]
    print('\nЗадание A2')
    [print(f'На кафедре {a} - {b} студентов') for a, b in A_2(one_to_many)]
    print('\nЗадание A3')
    [print(f'На кафедре {c}\t- {b} студентов. Староста {a}') for a, b, c in
A_3(many_to_many)]

if __name__ == '__main__':
    main()

```

```
"D:\Users\The Cat\PycharmProjects\schoo1\venv\Scripts\python.exe"
```

Задание A1

ИУ5 - Артамонов, 25 студентов

ИУ6 - Белодедов, 35 студентов

Задание A2

На кафедре ИУ5 - 25 студентов

На кафедре ИУ6 - 35 студентов

На кафедре РЛ6 - 105 студентов

Задание A3

На кафедре ИУ12 - 35 студентов. Староста Белодедов

На кафедре ИУ5 - 25 студентов. Староста Артамонов

На кафедре ИУ6 - 35 студентов. Староста Белодедов

На кафедре РЛ6 - 45 студентов. Староста Иваненко

На кафедре РЛ6 - 35 студентов. Староста Иванов

На кафедре РЛ6 - 25 студентов. Староста Иванин

На кафедре ФН4 - 25 студентов. Староста Артамонов

На кафедре ЗЗ - 45 студентов. Староста Иваненко

На кафедре ЗЗ - 35 студентов. Староста Иванов

На кафедре ЗЗ - 25 студентов. Староста Иванин

Process finished with exit code 0