

```
from
operator
import
itemgetter
```

```
class Emp:
    """Сотрудник"""
    def __init__(self, id, fio, sal, dep_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.dep_id = dep_id
```

```
class Dep:
    """Отдел"""
    def __init__(self, id, name):
        self.id = id
        self.name = name
```

```
class EmpDep:
    """
    '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 = [
    Dep(1, 'отдел кадров'),
    Dep(2, 'архивный отдел ресурсов'),
    Dep(3, 'бухгалтерия'),

    Dep(11, 'отдел (другой) кадров'),
    Dep(22, 'архивный (другой) отдел ресурсов'),
    Dep(33, '(другая) бухгалтерия'),

]
```

```
# Сотрудники
```

```
emps = [
    Emp(1, 'Артамонов', 25000, 1),
    Emp(2, 'Петров', 35000, 2),
    Emp(3, 'Иваненко', 45000, 3),
    Emp(4, 'Иванов', 35000, 3),
    Emp(5, 'Иванин', 25000, 3),
]
```

```
emps_deps = [
    EmpDep(1,1),
    EmpDep(2,2),
    EmpDep(3,3),
    EmpDep(3,4),
    EmpDep(3,5),

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

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

    # Соединение данных один-ко-многим
    one_to_many = [(e.fio, e.sal, 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.fio, e.sal, dep_name)
                    for dep_name, dep_id, emp_id in many_to_many_temp
                    for e in emps if e.id==emp_id]

    print('Задание A1')
```

```

res_11 = sorted(one_to_many, key=itemgetter(0))
print(res_11)

print('\nЗадание A2')
res_12_unsorted = []

for d in deps:
    d_emps = list(filter(lambda x: x[2] == d.name, one_to_many))
    if len(d_emps) > 0:
        l_count = len(d_emps)
        res_12_unsorted.append((d.name, l_count))
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание A3')
res_13 = {}

for e in emps:
    if 'ов' == e.fio[-2:]:

        d_emps = list(filter(lambda i: i[0]==e.fio, many_to_many))

        d_emps_names = [x[2] for x in d_emps]
        res_13[e.fio] = d_emps_names

print(res_13)

if __name__ == '__main__':
    main()

```

```

Задание A1
[('Артамонов', 25000, 'отдел кадров'), ('Иваненко', 45000, 'бухгалтерия'), ('Иванин', 25000, 'бухгалтерия'), ('Иванов', 35000, 'бухгалтерия'), ('Петров', 35000, 'архивный отдел ресурсов')]

Задание A2
[('бухгалтерия', 3), ('отдел кадров', 1), ('архивный отдел ресурсов', 1)]

Задание A3
{'Артамонов': ['отдел кадров', 'отдел (другой) кадров'], 'Петров': ['архивный отдел ресурсов', 'архивный (другой) отдел ресурсов'], 'Иванов': ['бухгалтерия', '(другая) бухгалтерия']}
Press any key to continue . . . █

```