

Вариант Г.

«Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

«Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.

«Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

30 Факультет Университет

```
from operator import itemgetter
```

```
class Факультет:
```

```
    """Факультет"""
```

```
    def __init__(self, id, название, зарплата_преподавателя, университет_id):
```

```
        self.id = id
```

```
        self.название = название
```

```
        self.зарплата_преподавателя = зарплата_преподавателя
```

```
        self.университет_id = университет_id
```

```
class Университет:
```

```
    """Университет"""
```

```
    def __init__(self, id, название):
```

```
        self.id = id
```

```
        self.название = название
```

```
class ФакультетыУниверситета:
```

```
    """
```

```
    'Факультеты университета' для реализации
```

```
    связи многие-ко-многим
```

```
    """
```

```
    def __init__(self, университет_id, факультет_id):
```

```
        self.университет_id = университет_id
```

```
        self.факультет_id = факультет_id
```

```
# Университеты
```

```
университеты = [
```

```
    Университет(1, 'МГУ'),
```

```
    Университет(2, 'СПбГУ'),
```

```
    Университет(3, 'МФТИ'),
```

```
    Университет(4, 'АГУ')
```

```
]
```

```
# Факультеты
```

```
факультеты = [
```

```
    Факультет(1, 'Математический', 50000, 1),
```

```
    Факультет(2, 'Физический', 60000, 1),
```

```
    Факультет(3, 'Исторический', 45000, 2),
```

```
    Факультет(4, 'Биологический', 55000, 3),
```

```
    Факультет(5, 'Астрономический', 70000, 4)
```

```
]
```

```
факультеты_университетов = [
```

```
    ФакультетыУниверситета(1, 1),
```

```
    ФакультетыУниверситета(1, 2),
```

```
    ФакультетыУниверситета(2, 3),
```

```
    ФакультетыУниверситета(3, 4),
```

```
    ФакультетыУниверситета(4, 5)
```

```
]
```

```

def main():
    one_to_many = [(f.название, f.зарплата_преподавателя, u.название)
                    for u in университеты
                    for f in факультеты
                    if f.университет_id == u.id]

    many_to_many_temp = [(u.название, fu.университет_id, fu.факультет_id)
                          for u in университеты
                          for fu in факультеты_университетов
                          if u.id == fu.университет_id]

    many_to_many = [(f.название, f.зарплата_преподавателя, университет_название)
                    for университет_название, университет_id, факультет_id in
many_to_many_temp
                    for f in факультеты if f.id == факультет_id]

    print('Задание Г1')
    res_Г1 = [item for item in one_to_many if item[2].startswith('А')]
    print(res_Г1)

    print('\nЗадание Г2')
    res_Г2_unsorted = []
    for u in университеты:
        u_факультеты = list(filter(lambda i: i[2] == u.название, one_to_many))
        if len(u_факультеты) > 0:
            u_зарплаты = [зарплата for _, зарплата, _ in u_факультеты]
            u_макс_зарплата = max(u_зарплаты)
            res_Г2_unsorted.append((u.название, u_макс_зарплата))
    res_Г2 = sorted(res_Г2_unsorted, key=itemgetter(1), reverse=True)
    print(res_Г2)

    print('\nЗадание Г3')
    res_Г3 = {}
    for u in университеты:
        u_факультеты = list(filter(lambda i: i[2] == u.название, many_to_many))
        u_факультеты_названия = [название for название, _, _ in u_факультеты]
        res_Г3[u.название] = u_факультеты_названия
    print(res_Г3)

if __name__ == '__main__':
    main()

```

Задание Г1

[('Астрономический', 70000, 'АГУ')]

Задание Г2

[('АГУ', 70000), ('МГУ', 60000), ('МФТИ', 55000), ('СПбГУ', 45000)]

Задание Г3

{'МГУ': ['Математический', 'Физический'], 'СПбГУ': ['Исторический'], 'МФТИ': ['Биологический'], 'АГУ': ['Астрономический']}