

**Условие:**

**Вариант Г. 26**

26	Студенческая группа	Учебный курс
----	---------------------	--------------

1. «Студенческая группа» и «Учебный курс» связаны соотношением один-ко-многим. Выведите список всех студенческих групп, у которых название начинается с буквы «А», и список учебного курса, в котором они работают.
2. «Студенческая группа» и «Учебный курс» связаны соотношением один-ко-многим. Выведите список учебных курсов с максимальной зарплатой студенческой группы в каждом учебном курсе, отсортированный по максимальной зарплате.
3. «Студенческая группа» и «Учебный курс» связаны соотношением многие-ко-многим. Выведите список всех связанных студенческих групп и учебного курса, отсортированный по студенческой группы, сортировка по учебному курсу произвольная.

```
#!/usr/bin/env python
```

```
from operator import itemgetter
```

```
class teachers:
```

```
    """Студенческая группа"""
```

```
    def __init__(self, id, name, price, trainingcourse_id):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.price = price
```

```
        self.trainingcourse_id = trainingcourse_id
```

```
class Trainingcourse:
```

```
    """Учебный курс"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
class teachersTrainingcourse:
```

```
    """Учебный курс к студенческой группы, для реализации многие-ко-многим"""
```

```
    def __init__(self, teachers_id, trainingcourse_id):
```

```
        self.teachers_id = teachers_id
```

```
        self.trainingcourse_id = trainingcourse_id
```

```

if __name__ == "__main__":
    Trainingcourse = [
        Trainingcourse(0, "Приборостроительный"),
        Trainingcourse(1, "Машиностроительные технологии"),
        Trainingcourse(2, "Разработка игр"),
        Trainingcourse(3, "Информационные технологии"),
        Trainingcourse(4, "Специальное машиностроение"),
        Trainingcourse(5, "Аэрокосмический"),
    ]

    teachers = [
        teachers(0, "ИУ5", 250000, 0),
        teachers(1, "ИУ1", 150000, 1),
        teachers(2, "ИУ2", 28000, 2),
        teachers(3, "МТ8", 164000, 3),
        teachers(4, "РК9", 20800, 3),
        teachers(5, "Э9", 1047000, 3),
        teachers(6, "ИМБ1", 110000, 3),
        teachers(7, "ИМБ3", 120000, 2),
        teachers(8, "ПС4", 1300, 2),
        teachers(9, "РТ5", 260000, 1),
        teachers(10, "СМ9", 3020, 4),
        teachers(11, "СМ11", 4080, 5),
        teachers(12, "СМ1", 72000, 5),
        teachers(13, "ПС2", 6040, 4),
        teachers(14, "СМ3", 7070, 3),
    ]

    teachers_Trainingcourse = [
        teachers_Trainingcourse(0, 0),
        teachers_Trainingcourse(1, 1),
        teachers_Trainingcourse(2, 2),
        teachers_Trainingcourse(3, 3),
        teachers_Trainingcourse(3, 4),
        teachers_Trainingcourse(3, 5),
        teachers_Trainingcourse(5, 5),
        teachers_Trainingcourse(4, 4),
        teachers_Trainingcourse(8, 1),
        teachers_Trainingcourse(11, 5),
    ]

    # Соединение данных один-ко-многим
    one_to_many = [(d.name, d.price, s.name)
                    for s in Trainingcourse
                    for d in teachers
                    if d.trainingcourse_id==s.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(s.name, ds.trainingcourse_id, ds.teachers_id)
                          for s in Trainingcourse

```

```

        for ds in teachers_Trainingcourse
        if s.id==ds.trainingcourse_id]
    many_to_many = [(d.name, d.price, Trainingcourse_name)
        for Trainingcourse_name, trainingcourse_id, teachers_id in
many_to_many_temp
        for d in teachers if d.id==teachers_id]

    print("Задание Г1")
    res_11 = {}
    selected_Trainingcourse = [one_traicour[2] for one_traicour in one_to_many if
one_traicour[2].startswith('a') or one_traicour[2].startswith('A')]
    for Trainingcourse_name in selected_Trainingcourse:
        teachers_for_traicour = [(one_teachers[0],one_teachers[1]) for
one_teachers in one_to_many if one_teachers[2]==Trainingcourse_name]
        res_11.update({Trainingcourse_name:teachers_for_traicour})
    print(res_11)
    print()

    print("Задание Г2")
    res_12_unsorted = []
    for s in Trainingcourse:
        s_teachers = list(filter(lambda i: i[2]==s.name, one_to_many))
        if len(s_teachers) > 0:
            s_prices = [price for _,price,_ in s_teachers]
            s_price_max = max(s_prices)
            res_12_unsorted.append((s.name, s_price_max))

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

    print("Задание Г3")
    res_13 = {}
    Trainingcourse.sort(key=lambda one_Trainingcourse: one_Trainingcourse.name)
    for s in Trainingcourse:
        s_teachers = list(filter(lambda i: i[2]==s.name, many_to_many))
        s_teachers_names = [x for x,_,_ in s_teachers]
        res_13[s.name] = s_teachers_names
    print(res_13)

```

PS C:\Users\anzhe\OneDrive\Рабочий стол\ПК1> & 'C:\Users\anzhe\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\anzhe\.vscode\extensions\ms-python.python-2021.10.1336267007\pythonFiles\lib\python\debugpy\launcher' '51077' '--' 'c:\Users\anzhe\OneDrive\Рабочий стол\ПК1\rk1\_Semenova.py'

Задание Г1  
{'Аэрокосмический': [('СМ11', 4080), ('СМ1', 72000)]}

Задание Г2  
[('Информационные технологии', 1047000), ('Машиностроительные технологии', 260000), ('Приборостроительный', 250000), ('Разработка игр', 120000), ('Аэрокосмический', 72000), ('Специальное машиностроение', 6040)]

Задание Г3  
{'Аэрокосмический': ['МТ8', 'Э9', 'СМ11'], 'Информационные технологии': ['МТ8'], 'Машиностроительные технологии': ['ИУ1', 'ПС4'], 'Приборостроительный': ['ИУ5'], 'Разработка игр': ['ИУ2'], 'Специальное машиностроение': ['МТ8', 'РК9']}  
PS C:\Users\anzhe\OneDrive\Рабочий стол\ПК1>