

**Текст программы:**

```
from operator import itemgetter

class pk:
    """Компьютер"""
    def __init__(self, id, processor, ram, classroom_id):
        self.id = id
        self.processor = processor
        self.ram = ram
        self.classroom_id = classroom_id

class classroom:
    """Аудитория"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class classroom_pk:
    """Компьютер в аудитории(для связи многие ко многим)"""
    def __init__(self, classroom_id, pk_id):
        self.pk_id = pk_id
        self.classroom_id = classroom_id

#Аудитории
classrooms = [
    classroom(1, "Дисплейный класс физики"),
    classroom(2, "Дисплейный класс информационного направления"),
    classroom(3, "Дисплейный класс химии"),
    classroom(4, "Дисплейный класс математического направления"),
    classroom(5, "Дисплейный класс географии"),
    classroom(6, "Дисплейный класс направления информационной безопасности"),
]

#Компьютеры
pks = [
    pk(1, "Intel i5", 8, 1),
    pk(2, "Intel i7", 16, 2),
    pk(3, "Intel i9", 16, 3),
    pk(4, "Ryzen 5", 32, 3),
    pk(5, "Ryzen 7", 8, 3),
]

classrooms_pks = [
    classroom_pk(1,1),
    classroom_pk(2,2),
    classroom_pk(3,3),
```

```

classroom_pk(3,4),
classroom_pk(3,5),
classroom_pk(4,1),
classroom_pk(5,2),
classroom_pk(6,3),
classroom_pk(6,4),
classroom_pk(6,5),
]

def main():
    one_to_many = [(p.processor, p.ram, c.name)
                    for c in classrooms
                    for p in pks
                    if p.classroom_id==c.id]

    many_to_many_temp = [(c.name, p.classroom_id, p.pk_id)
                          for c in classrooms
                          for p in classrooms_pks
                          if c.id==p.classroom_id]

    many_to_many = [(p.processor, p.ram, classroom_name)
                     for classroom_name, classroom_id, pk_id in many_to_many_temp
                     for p in pks if p.id==pk_id]

    #Сортировка по аудиториям
    print('Задание A1')
    res_1 = sorted(one_to_many, key=itemgetter(2))
    print(res_1)

    #Сортировка аудиторий по суммарной ram
    print('\nЗадание A2')
    res_2 = []
    for cl in classrooms:
        temp_class = list(filter(lambda i: i[2] == cl.name, one_to_many))
        total_ram=0
        if len(temp_class) > 0:
            for i in temp_class:
                total_ram+=i[1]
            res_2.append((cl.name, total_ram))

    res_2 = sorted(res_2, key=itemgetter(1))
    print(res_2)

    #Список всех аудиторий, в названии которых есть слово "направления" с процес-
    #сорами компьютеров в них
    print('\nЗадание A3')
    res_3 = {}
    for cl in classrooms:
        if 'направления' in cl.name:
            this_class=list(filter(lambda i: i[2] == cl.name, many_to_many))
            proc_list = [i[0] for i in this_class]

```

```

        res_3[cl.name]=proc_list
    print(res_3)

if __name__ == '__main__':
    main()

```

**Результат выполнения:**

**Задание A1**

**[('Intel i7', 16, 'Дисплейный класс информационного направления'), ('Intel i5', 8, 'Дисплейный класс физики'), ('Intel i9', 16, 'Дисплейный класс химии'), ('Ryzen 5', 32, 'Дисплейный класс химии'), ('Ryzen 7', 8, 'Дисплейный класс химии')]**

**Задание A2**

**[('Дисплейный класс физики', 8), ('Дисплейный класс информационного направления', 16), ('Дисплейный класс химии', 56)]**

**Задание A3**

**{'Дисплейный класс информационного направления': ['Intel i7'], 'Дисплейный класс математического направления': ['Intel i5'], 'Дисплейный класс направления информационной безопасности': ['Intel i9', 'Ryzen 5', 'Ryzen 7']}**

```

Задание A1
[('Intel i7', 16, 'Дисплейный класс информационного направления'), ('Intel i5', 8, 'Дисплейный класс физики'), ('Intel i9', 16, 'Дисплейный класс химии'), ('Ryzen 5', 32, 'Дисплейный класс химии'), ('Ryzen 7', 8, 'Дисплейный класс химии')]

Задание A2
[('Дисплейный класс физики', 8), ('Дисплейный класс информационного направления', 16), ('Дисплейный класс химии', 56)]

Задание A3
{'Дисплейный класс информационного направления': ['Intel i7'], 'Дисплейный класс математического направления': ['Intel i5'], 'Дисплейный класс направления информационной безопасности': ['Intel i9', 'Ryzen 5', 'Ryzen 7']}

```