

```
from operator import itemgetter
```

```
class Hrd:
```

```
    """Жесткий диск"""
```

```
    def __init__(self, id, memory, type, name, cmp_id):
```

```
        self.id = id
```

```
        self.memory = memory
```

```
        self.type = type
```

```
        self.name = name
```

```
        self.cmp_id = cmp_id
```

```
class Cmp:
```

```
    """Компьютер"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
        self.type = type
```

```
        self.os = os
```

```
class CmpHrd:
```

```
    """
```

```
    'Жесткие диски компьютера' для реализации  
связи многие-ко-многим
```

```
    """
```

```
    def __init__(self, cmp_id, hrd_id):
```

```
        self.cmp_id = cmp_id
```

```
        self.hrd_id = hrd_id
```

```
# Компьютеры
```

```
cmps = [
```

```
    Cmp(1, "Рабочая станция 1", "настольный", "Windows 10"),
```

```
    Cmp(2, "Ноутбук 2", "ноутбук", "Ubuntu 20.04"),
```

```
    Cmp(3, "Сервер 3", "сервер", "Windows Server 2019"),
```

```
    Cmp(4, "Игровой ПК", "настольный", "Windows 11"),
```

```
    Cmp(5, "Лаптоп 4", "ноутбук", "macOS Monterey"),
```

```
]
```

```
# Жесткие диски
```

```
hrds = [
```

```
    Hrd(1, 256, "SSD", "Samsung", 1),
```

```
    Hrd(2, 512, "HDD", "Lacie", 2),
```

```
    Hrd(3, 1024, "SSD", "Toshi", 3),
```

```
    Hrd(4, 2000, "HDD", "Fujitsu", 3),
```

```
Hrd(5, 4000, "SSD", "BMSTU Special", 3),  
]
```

```
cmps_hrds = [  
    CmpHrd(1, 1),  
    CmpHrd(1, 2),  
    CmpHrd(2, 1),  
    CmpHrd(2, 2),  
    CmpHrd(3, 1),  
    CmpHrd(3, 3),  
    CmpHrd(4, 4),  
    CmpHrd(5, 5),  
]
```

```
def main():  
    """Основная функция"""  
  
    # Соединение данных один-ко-многим  
    one_to_many = [(h.name, h.memory, c.name)  
        for h in hrds  
        for c in cmps  
        if h.cmp_id == c.id]  
  
    # Соединение данных многие-ко-многим  
    many_to_many_temp = [(c.name, ch.cmp_id, ch.hrd_id)  
        for c in cmps  
        for ch in cmps_hrds  
        if c.id == ch.cmp_id]  
  
    many_to_many = [(h.name, h.memory, cmp_name)  
        for cmp_name, cmp_id, hrd_id in many_to_many_temp  
        for h in hrds if h.id == hrd_id]  
    print('Задание A1')  
    res_11 = sorted(one_to_many, key=itemgetter(2))  
    print(res_11)  
  
    print('\nЗадание A2')  
    res_12_unsorted = []  
    # Перебираем все компьютеры  
    for c in cmps:  
        # Список жестких дисков компьютера  
        c_hrds = list(filter(lambda i: i[2] == c.name, one_to_many))  
        # Если в компьютере есть жесткий диск  
        if len(c_hrds) > 0:  
            # Памяти жестких дисков компьютера  
            c_memorys = [memory for _, memory, _ in c_hrds]  
            # Суммарная память всех жестких дисков компьютера  
            c_memorys_sum = sum(c_memorys)  
            res_12_unsorted.append((c.name, c_memorys_sum))
```

```
# Сортировка по суммарной памяти
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)
```

```
print('\nЗадание А3')
res_13 = {}
# Перебираем все компьютеры
for c in cmprs:
    # Список жестких дисков компьютера
    c_hrds = list(filter(lambda i: i[2] == c.name, many_to_many))
    # Только название жесткого диска
    c_hrds_names = [x for x, _, _ in c_hrds]
    # Добавляем результат в словарь
    # ключ - компьютер, значение - список жестких дисков
    res_13[c.name] = c_hrds_names

print(res_13)
```

```
if __name__ == '__main__':
    main()
```

Результаты работы:

Задание А1

```
[('Lacie', 512, 'Ноутбук 2'), ('Samsung', 256, 'Рабочая станция 1'), ('Toshi', 1024, 'Сервер 3'),
('Fujitsu', 2000, 'Сервер 3'), ('BMSTU Special', 4000, 'Сервер 3')]
```

Задание А2

```
[('Сервер 3', 7024), ('Ноутбук 2', 512), ('Рабочая станция 1', 256)]
```

Задание А3

```
{'Рабочая станция 1': ['Samsung', 'Lacie'], 'Ноутбук 2': ['Samsung', 'Lacie'], 'Сервер 3': ['Samsung',
'Toshi'], 'Игровой ПК': ['Fujitsu'], 'Лaptop 4': ['BMSTU Special']}
```