

```

from operator import itemgetter
class driver:
    def __init__(self, id, fio, sal, carPark_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.carPark_id = carPark_id
class carPark:
    def __init__(self, id, name):
        self.id = id
        self.name = name
class driverCarPark:
    def __init__(self, carPark_id, driver_id):
        self.carPark_id = carPark_id
        self.driver_id = driver_id
car_parks = [
    carPark(1, 'Машинка'),
    carPark(2, 'Скорость'),
    carPark(3, 'Автоцентр'),
    carPark(11, 'Need for speed'),
    carPark(22, 'Быстрее ветра'),
    carPark(33, 'АвтоБас'),
]
drivers = [
    driver(1, 'Артамонов', 25000, 1),
    driver(2, 'Петров', 35000, 2),
    driver(3, 'Иваненко', 45000, 3),
    driver(4, 'Акимов', 35000, 3),
    driver(5, 'Иванин', 30000, 3),
    driver(6, 'Абдуллаев', 50000, 1),
    driver(7, 'Алексеев', 38000, 2),
]
driver_car = [
    driverCarPark(1, 1),
    driverCarPark(2, 2),
    driverCarPark(3, 3),
    driverCarPark(3, 4),
    driverCarPark(3, 5),
    driverCarPark(2, 6),
    driverCarPark(1, 7),
    driverCarPark(11, 1),
    driverCarPark(22, 2),
    driverCarPark(33, 3),
    driverCarPark(33, 4),
    driverCarPark(33, 5),
    driverCarPark(22, 6),
    driverCarPark(11, 7),
]
def main():
    one_to_many = [(e.fio, e.sal, d.name)
                    for d in car_parks
                    for e in drivers
                    if e.carPark_id == d.id]
    many_to_many_temp = [(d.name, ed.carPark_id, ed.driver_id)
                          for d in car_parks
                          for ed in driver_car
                          if d.id == ed.carPark_id]
    many_to_many = [(e.fio, e.sal, car_park_name)
                     for car_park_name, car_park_id, driver_id in many_to_many_temp
                     for e in drivers if e.id == driver_id]
    print('Задание A1')
    res_11 = {}

```

```

for d in drivers:
    for e in car_parks:
        if 'A' in d.fio:
            d_parks = list(filter(lambda i: i[0] == d.fio, one_to_many))
            d_parks_names = [x for _, x in d_parks]
            res_11[d.fio] = d_parks_names

print(res_11)
print("\nЗадание A2")
res_12_unsorted = []
for d in car_parks:
    d_driver = list(filter(lambda i: i[2] == d.name, one_to_many))
    if len(d_driver) > 0:
        d_sals = [sal for _, sal, _ in d_driver]
        d_sals_min = min(d_sals)
        res_12_unsorted.append((d.name, d_sals_min))
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)
print("\nЗадание A3")
res_13 = sorted(many_to_many, key=itemgetter(0))
print(res_13)
if __name__ == '__main__':
    main()

```



