```python
from operator import itemgetter
from collections import defaultdict

class Conductor:
    def __init__(self, id, name, salary):
        self.id = id
        self.name = name
        self.salary = salary

class Orchestra:
    def __init__(self, id, name, conductor_id):
        self.id = id
        self.name = name
        self.conductor_id = conductor_id

class ConductorOrchestra:
    def __init__(self, conductor_id, orchestra_id):
        self.conductor_id = conductor_id
        self.orchestra_id = orchestra_id

def main():
    one_to_many = [(c.name, c.salary, o.name) for o in orchestras for c in
conductors if o.conductor_id == c.id]
    print('Задание 1')
    res_A1 = sorted(one_to_many, key=itemgetter(2))
    print(res_A1)

    print('\n')
    print('Задание 2')
    res_A2_unsorted = [(orchestra.name, sum(c.salary for c in conductors if
c.id in [co.conductor_id for co in conductor_orchestras if co.orchestra_id ==
orchestra.id]))
                       for orchestra in orchestras]
    res_A2 = sorted(res_A2_unsorted, key=itemgetter(1), reverse=True)
    print(res_A2)
    print('\n')
    print('Задание 3')
    res_A3 = defaultdict(list)
    for orchestra in orchestras:
        if orchestra.name.startswith('Symphony'):
            conductor_ids = [co.conductor_id for co in conductor_orchestras
if co.orchestra_id == orchestra.id]
            conductor_names = [c.name for c in conductors if c.id in
conductor_ids]
            res_A3[orchestra.name].extend(conductor_names)
    print(res_A3)

if __name__ == '__main__':
    conductors = [
        Conductor(1, 'John Smith', 5000),
        Conductor(2, 'Emily Johnson', 6000),
        Conductor(3, 'Michael Davis', 5500)
    ]
    orchestras = [
        Orchestra(1, 'Symphony Orchestra', 1),
        Orchestra(2, 'Chamber Orchestra', 2),
        Orchestra(3, 'Philharmonic Orchestra', 3)
    ]
    conductor_orchestras = [
        ConductorOrchestra(1, 1),
        ConductorOrchestra(2, 2),
        ConductorOrchestra(3, 3),
        ConductorOrchestra(1, 2),
        ConductorOrchestra(2, 1),
```

```
        ConductorOrchestra(3, 2),
    ]
    main()
```

Results

Задание 1

[('Emily Johnson', 6000, 'Chamber Orchestra'), ('Michael Davis', 5500, 'Philharmonic Orchestra'), ('John Smith', 5000, 'Symphony Orchestra')]

Задание 2

[('Chamber Orchestra', 16500), ('Symphony Orchestra', 11000), ('Philharmonic Orchestra', 5500)]

Задание 3

defaultdict(<class 'list'>, {'Symphony Orchestra': ['John Smith', 'Emily Johnson']})

Process finished with exit code 0