Код

```python
# вариант запроса Д
# вариант предметной области 11 : программа - компьютер
from operator import itemgetter


class Program:
    def __init__(self, id, name, user, orchestra_id):
        self.id = id
        self.name = name
        self.user = user
        self.orchestra_id = orchestra_id


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


class CompProg:
    def __init__(self,  comp_id, prog_id):
        self.comp_id = comp_id
        self.prog_id = prog_id

comp = [
    Computer(1, "PC1"),
    Computer(2, "PC2"),
    Computer(3, "PC3"),
    Computer(4, "PC4"),
    Computer(5, "PC5"),
    Computer(6, "PC6")
]

prog = [
    Program(1, "Visual Studion", 10000, 1),
    Program(2, "CLion", 22000, 2),
    Program(3, "PyCharm CE", 45500, 2),
    Program(4, "Xcode", 52000, 3),
    Program(5, "WebSton", 15000, 3),
    Program(6, "Atom", 100000, 3),
    Program(7, "VirtualBon", 10400, 3)
]

comp_prog = [
    CompProg(1, 1),
    CompProg(2, 2),
    CompProg(2, 3),
    CompProg(3, 4),
    CompProg(3, 5),
    CompProg(3, 6),
    CompProg(3, 7),
    CompProg(4, 1),
    CompProg(5, 2),
    CompProg(5, 3),
    CompProg(6, 4),
    CompProg(6, 5),
    CompProg(6, 6),
    CompProg(6, 7),
]
```

```
def main():
    one_to_many = [(c.name, c.user, o.name)
                   for o in comp
                   for c in prog
                   if c.orchestra_id == o.id]

    many_to_many_temp = [(o.name, co.comp_id, co.prog_id)
                         for o in comp
                         for co in comp_prog
                         if o.id == co.comp_id]

    many_to_many = [(c.name, c.user, orch_name)
                    for orch_name, comp_id, prog_id in many_to_many_temp
                    for c in prog if c.id == prog_id]

    print('Задание Д1')
    res1 = []
    for i in one_to_many:
        if i[0][-2:] == "on":
            res1.append(i[0:3:2])
    print(res1)

    print('\nЗадание Д2')
    res2_unsorted = []
    for o in comp:
        o_comps = list(filter(lambda i: i[2] == o.name, one_to_many))
        if len(o_comps) > 0:
            o_user = [listeners for _, listeners, _ in o_comps]
            o_user_sum = sum(o_user)
            o_user_count = len(o_user)
            o_user_average = o_user_sum / o_user_count
            res2_unsorted.append((o.name, int(o_user_average)))
    res2 = sorted(res2_unsorted, key=itemgetter(1), reverse=True)
    print(res2)

    print('\nЗадание Д3')
    res3 = {}
    for o in comp:
        if o.name[0] == "P":
            o_comps = list(filter(lambda i: i[2] == o.name, many_to_many))
            o_comps_user = [x for x, _, _ in o_comps]
            res3[o.name] = o_comps_user
    print(res3)


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

Результат

Задание Д1

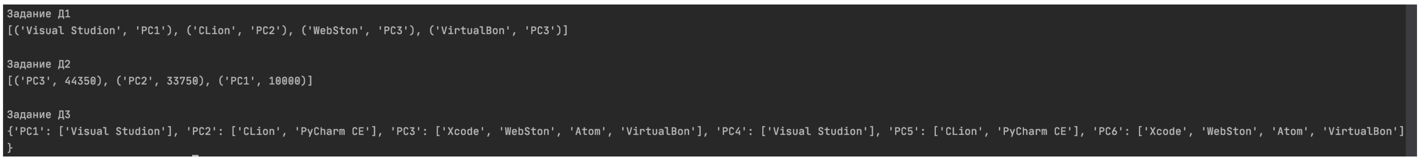
[('Visual Studion', 'PC1'), ('CLion', 'PC2'), ('WebSton', 'PC3'), ('VirtualBon', 'PC3')]

Задание Д2

[('PC3', 44350), ('PC2', 33750), ('PC1', 10000)]

Задание Д3

{'PC1': ['Visual Studion'], 'PC2': ['CLion', 'PyCharm CE'], 'PC3': ['Xcode', 'WebSton', 'Atom', 'VirtualBon'], 'PC4': ['Visual Studion'], 'PC5': ['CLion', 'PyCharm CE'], 'PC6': ['Xcode', 'WebSton', 'Atom', 'VirtualBon']}



```
Задание Д1
[('Visual Studion', 'PC1'), ('CLion', 'PC2'), ('WebSton', 'PC3'), ('VirtualBon', 'PC3')]

Задание Д2
[('PC3', 44350), ('PC2', 33750), ('PC1', 10000)]

Задание Д3
{'PC1': ['Visual Studion'], 'PC2': ['CLion', 'PyCharm CE'], 'PC3': ['Xcode', 'WebSton', 'Atom', 'VirtualBon'], 'PC4': ['Visual Studion'], 'PC5': ['CLion', 'PyCharm CE'], 'PC6': ['Xcode', 'WebSton', 'Atom', 'VirtualBon']
}
```