

```

#Вариант запроса Д
#Вариант ПО 9 - Связь ОС - Компьютер
# выполнено Кожуро В.Е.
import math
import operator

class OS:
    # ОС
    def __init__(self, id, name, version):
        self.id = id
        self.name = name
        self.version = version

class PC:
    # ПК
    def __init__(self, id, name, cost, usedOS):
        self.id = id
        self.name = name
        self.cost = cost
        self.usedOS = usedOS

# класс СВЯЗИ
class Link:
    def __init__(self, id_OS, id_PC):
        self.id_OS = id_OS
        self.id_PC = id_PC

#ОС:
OSs = [
    OS(1, 'Windows', 10),
    OS(2, 'Linux Ubuntu', 12.3),
    OS(3, 'Linux Mint', 20),
    OS(4, 'MacOS', 12.15)
]

#ПК:
PCs = [
    PC(1, 'Acer1', 2300, 1),
    PC(2, 'Acer2', 2000, 2),
    PC(3, 'HP1', 1500, 2),
    PC(4, 'HP2', 1300, 3),
    PC(5, 'MacBook', 3000, 4),
    PC(6, 'Dell1', 2000, 1),
    PC(7, 'Dell2', 2500, 1),
    PC(8, 'HP3', 1500, 2),
    PC(9, 'iMac', 4000, 4),
]

# СВЯЗИ ОДИН-КО-МНОГИМ
Usable_OS = [
    Link(2, 1),
    Link(1, 2),
    Link(1, 3),
    Link(1, 4),
    Link(1, 8),
    Link(2, 4),
    Link(2, 6),
    Link(2, 7),
    Link(2, 8)
]

def main():
    #создание массива связей 1 к М
    OS_to_PC = [(pc.name, pc.cost, os.name)
                 for os in OSs
                 for pc in PCs
                 if pc.usedOS == os.id]

    #Создание обратных связей ПК к ОС:
    #сначала создаем связи ПК к ОС

```

```

PC_to_OS_unique = [(lk.id_PC, lk.id_OS, os.name)
                    for os in OSs
                    for lk in Usable_OS
                    if os.id == lk.id_OS]
#Добавляем связи ОС к ПК, т.к. они стали двунаправленными
PC_to_OS = [(pc.name, pc.cost, os_name)
             for pc_id, os_id, os_name in PC_to_OS_unique
             for pc in PCs
             if pc.id == pc_id]
PC_to_OS.extend(OS_to_PC)

print('Задание Д1')
res1 = []
for k in OS_to_PC:
    if k[0][-2:] == 'P1':
        res1.append(k[0:3:2])
print(res1)

print('\nЗадание Д2')
res2 = []
for os in OSs:
    os_pcs = list(filter(lambda i: i[2] == os.name, OS_to_PC))
    if len(os_pcs) > 0:
        os_cost = [cost for _, cost, _ in os_pcs]
        os_cost_sum = sum(os_cost)
        #Делим сумму на количество записей и округляем вниз
        res2.append((os.name, math.floor(os_cost_sum/len(os_cost))))

res2_sorted = sorted(res2, key=operator.itemgetter(1), reverse=True)
print(res2_sorted)

print('\nЗадание Д3')
res3 = {}
for k in OSs:
    if k.name[0] == 'L':
        M = list(filter(lambda i: i[2] == k.name, PC_to_OS))
        Names = [name for name, _, _ in M]
        res3[k.name] = Names
print(res3)

if __name__ == '__main__':
    main()

```

Результаты:

Задание Д1

[('HP1', 'Linux Ubuntu')]

Задание Д2

[('MacOS', 3500), ('Windows', 2266), ('Linux Ubuntu', 1666), ('Linux Mint', 1300)]

Задание Д3

{'Linux Ubuntu': ['Acer1', 'HP2', 'Dell1', 'Dell2', 'HP3', 'Acer2', 'HP1', 'HP3'], 'Linux Mint': ['HP2']}