

```
class OpSys:
    def __init__(self, id, name, price, comp_id):
        self.id = id
        self.name = name
        self.price = price
        self.comp_id = comp_id
```

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

```
class OsComp:
    def __init__(self, comp_id, os_id):
        self.comp_id = comp_id
        self.os_id = os_id
```

```
os_ = [
    OpSys(1, 'Windows 7 Home', 1000, 1),
    OpSys(2, 'Windows 7 Pro', 2000, 2),
    OpSys(3, 'Windows 8.1 Pro', 3000, 4),
    OpSys(4, 'Windows 10 Home', 4000, 3),
    OpSys(5, 'Windows 10 Pro', 11000, 4),
    OpSys(6, 'Arch Linux OS', 0, 2)
]
```

```
# Компьютеры
```

```
comps = [
    Comp(1, "Компьютер пользователя 1"),
```

```
Comp(2, "Компьютер администратора 2"),  
Comp(3, "Компьютер администратора 3"),  
Comp(4, "Компьютер общий 4")  
]
```

```
os_comps = [  
    OsComp(1, 1),  
    OsComp(1, 2),  
    OsComp(2, 2),  
    OsComp(3, 2),  
    OsComp(3, 3),  
    OsComp(4, 4),  
    OsComp(4, 5),  
    OsComp(5, 6)  
]
```

```
def main():  
    one_to_many = [(o.name, o.price, c.name)  
                    for c in comps  
                    for o in os_ if o.comp_id == c.id]  
  
    # Вывести компьютеры администратора и установленные на них OS  
    print('Задание A1:')  
    for os_name, price, comp_name in one_to_many:  
        if "администратора" in comp_name.lower():  
            print("{} с установленной OS {}".format(comp_name, os_name))  
    print()  
  
    # Компьютеры, отсортированные по среднему значению цены установленных ос  
    print("Задание A2:")  
    result = {}
```

```
for os, price, comp in one_to_many:
```

```
    if comp not in result:
```

```
        result[comp] = [price]
```

```
    else:
```

```
        result[comp].append(price)
```

```
result = {k: round(sum(v)/len(v), 2) for k, v, in result.items()}
```

```
for name, avg in sorted(result.items(), key = lambda i: i[1]):
```

```
    print('{} со средней ценой установленных OS {}'.format(name, avg))
```

```
print()
```

```
# Реализация связи многие-ко-многим
```

```
many_to_many_temp = [(c.name, oc.comp_id, oc.os_id)
```

```
                      for c in comps
```

```
                      for oc in os_comps
```

```
                      if c.id == oc.comp_id]
```

```
many_to_many = [(o.name, o.price, name)
```

```
                for name, comp_id, os_id in many_to_many_temp
```

```
                for o in os_
```

```
                if o.id == os_id]
```

```
print(many_to_many)
```

```
print()
```

```
# Выборка операционных систем со словом 'Pro' и список компьютеров,
```

```
# на которых установлены эти ОС
```

```
print("Задание A3:")
```

```
out = {}
```

```
for os, price, computer in many_to_many:
```

```
    if 'Pro' in os:
```

```
        if os not in out:
```

```

        out[os] = [computer]

    else:

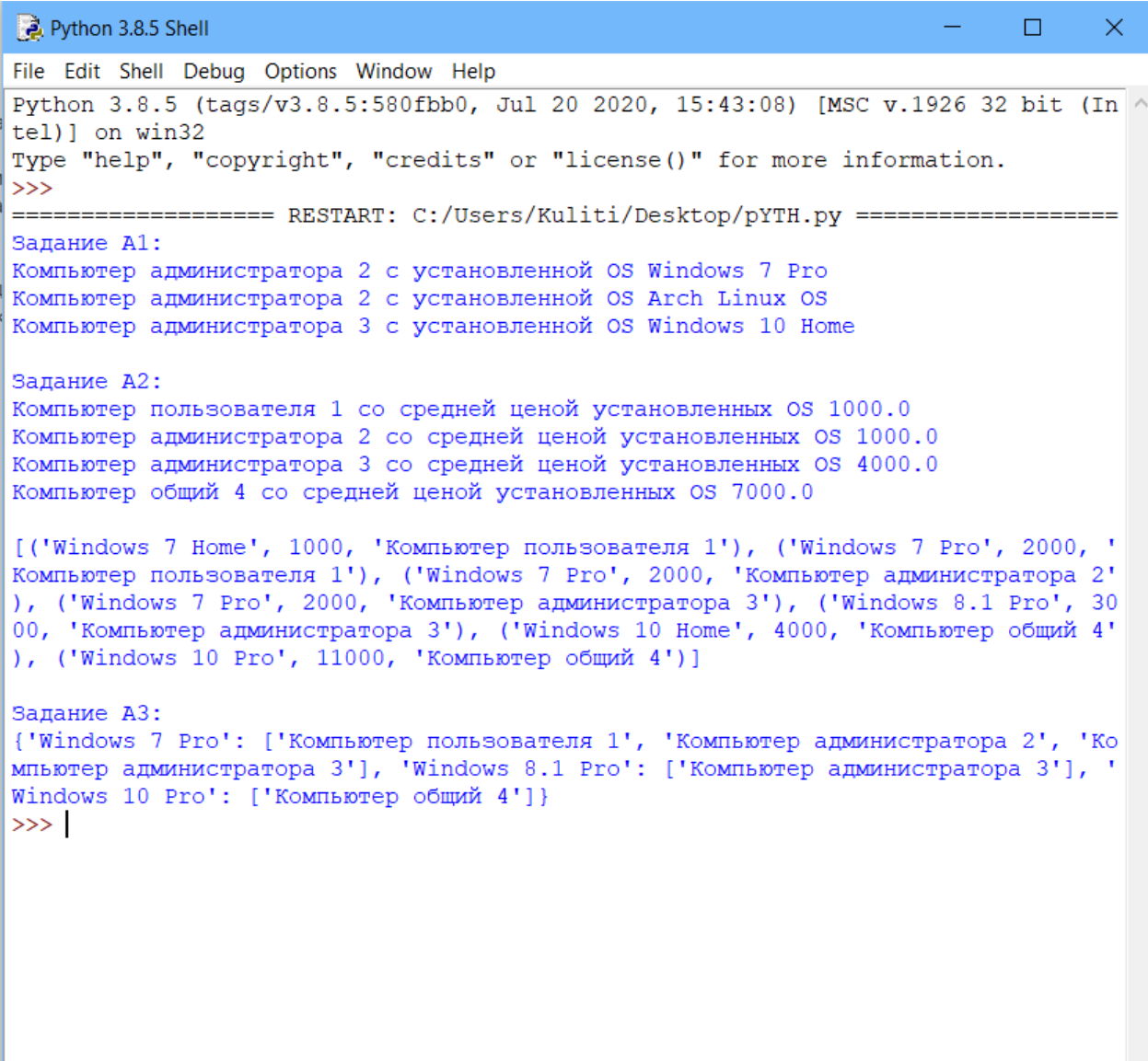
        out[os].append(computer)

print(out)

if __name__ == '__main__':

    main()

```



```

Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Kuliti/Desktop/pYTH.py =====
Задание A1:
Компьютер администратора 2 с установленной OS Windows 7 Pro
Компьютер администратора 2 с установленной OS Arch Linux OS
Компьютер администратора 3 с установленной OS Windows 10 Home

Задание A2:
Компьютер пользователя 1 со средней ценой установленных OS 1000.0
Компьютер администратора 2 со средней ценой установленных OS 1000.0
Компьютер администратора 3 со средней ценой установленных OS 4000.0
Компьютер общий 4 со средней ценой установленных OS 7000.0

[('Windows 7 Home', 1000, 'Компьютер пользователя 1'), ('Windows 7 Pro', 2000, 'Компьютер пользователя 1'), ('Windows 7 Pro', 2000, 'Компьютер администратора 2'), ('Windows 7 Pro', 2000, 'Компьютер администратора 3'), ('Windows 8.1 Pro', 3000, 'Компьютер администратора 3'), ('Windows 10 Home', 4000, 'Компьютер общий 4'), ('Windows 10 Pro', 11000, 'Компьютер общий 4')]

Задание A3:
{'Windows 7 Pro': ['Компьютер пользователя 1', 'Компьютер администратора 2', 'Компьютер администратора 3'], 'Windows 8.1 Pro': ['Компьютер администратора 3'], 'Windows 10 Pro': ['Компьютер общий 4']}
>>> |

```