

```
from operator import itemgetter
```

```
class brau:
```

```
    def __init__(self, id, name, size, comp_id):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.size = size
```

```
        self.comp_id = comp_id
```

```
class comp:
```

```
    def __init__(self, brau_id, name):
```

```
        self.id = brau_id
```

```
        self.name = name
```

```
class combra:
```

```
    def __init__(self, comp_id, brau_id):
```

```
        self.comp_id = comp_id
```

```
        self.brau_id = brau_id
```

```
computers = [
```

```
    comp(1, 'macintosh'),
```

```
    comp(2, 'dell'),
```

```
    comp(3, 'ibm'),
```

```
    comp(4, 'toshiba'),
```

```
    comp(5, 'msi')
```

```
]
```

```
brausers = [
```

```
    brau(1, 'safari', 173, 1),
```

```
    brau(2, 'chrome', 140, 2),
```

```
    brau(3, 'mozilla', 97, 3),
```

```
    brau(4, 'Opera', 201, 4),
```

```
    brau(5, 'Edge', 50, 5)
]
```

```
comps_braus = [
    combra(1, 1),
    combra(1, 2),
    combra(2, 2),
    combra(2, 4),
    combra(3, 3),
    combra(4, 2),
    combra(4, 4),
    combra(5, 2),
    combra(5, 3),
    combra(5, 4)
]
```

```
def main():
    one_to_many = [(b.name, b.size, c.name)
                    for b in brausers
                    for c in computers
                    if b.id == c.id
                    ]
    many_to_many_temp = [(c.name, cb.comp_id, cb.brau_id)
                          for c in computers
                          for cb in comps_braus
                          if c.id == cb.comp_id
                          ]
    many_to_many = [(b.name, b.size, comp_name)
                    for comp_name, comp_id, brau_id in many_to_many_temp
                    for b in brausers if b.id == brau_id
                    ]
```

```
print('Задание Д1')
res_1 = []
for brau in one_to_many:
    if brau[0][-1] == 'a':
        res_1.append((brau[0], brau[2]))
print(res_1)
```

```
print('Задание Д2')
res_2_unsorted = []
for c in computers:
    c_braus = list(filter(lambda i: i[2]==c.name, many_to_many))
    if len(c_braus) > 0:
        c_sizes = [size for _,size,_ in c_braus]
        c_sizes_sum = sum(c_sizes)
        k = len(c_sizes)
        res_2_unsorted.append((c.name, c_sizes_sum/k))
res_2 = sorted(res_2_unsorted, key=itemgetter(1), reverse=True)
print(res_2)
```

```
print('Задание Д3')
res_3 = {}
for c in computers:
    if 'msi' in c.name:
        c_braus = list(filter(lambda i: i[2]==c.name, many_to_many))
        c_braus_names = [x for x,_,_ in c_braus]
        res_3[c.name] = c_braus_names
print(res_3)
if __name__ == '__main__':
    main()
```

ПРИМЕР ВЫПОЛНЕНИЯ ПРОГРАММЫ:

```
Задание Д1  
[('mozilla', 'ibm'), ('opera', 'toshiba')]  
Задание Д2  
[('dell', 170.5), ('toshiba', 170.5), ('macintosh', 156.5), ('msi', 146.0), ('ibm', 97.0)]  
Задание Д3  
{'msi': ['chrome', 'mozilla', 'opera']}
```

Задание Д1

[('mozilla', 'ibm'), ('opera', 'toshiba')]

Задание Д2

[('dell', 170.5), ('toshiba', 170.5), ('macintosh', 156.5), ('msi', 146.0), ('ibm', 97.0)]

Задание Д3

{'msi': ['chrome', 'mozilla', 'opera']}