```
from datetime import date


class File:
    def __init__(self, id, name, creation_date, modification_date, author,
size, filedir_id):
        self.id = id
        self.name = name
        self.creation_date = creation_date
        self.modification_date = modification_date
        self.author = author
        self.size = size
        self.filedir_id = filedir_id

    def __repr__(self):
        return self.name


class FileDirectory:
    def __init__(self, id, path):
        self.id = id
        self.path = path

    def __repr__(self):
        return self.path


class FileFileDirectory:
    def __init__(self, file_id, file_directory_id):
        self.file_id = file_id
        self.file_directory_id = file_directory_id


filedirs = [FileDirectory(1, '/root/PyCharm Projects/lab1'),
            FileDirectory(2, '/root/PyCharm Projects/lab2'),
            FileDirectory(3, '/root/PyCharm Projects/lab3'),
            FileDirectory(4, '/root/Desktop/')]

files = [File(1, 'lab2.py', date(2023, 9, 23), date(2023, 9, 23),
'rasulov1337', 1024, 2),
         File(2, 'lab1.py', date(2023, 10, 19), date(2023, 10, 19),
'rasulov1337', 10, 1),
         File(3, 'readme.txt', date(2023, 10, 1), date(2023, 10, 15), 'root',
10, 3),
         File(4, '.gitignore', date(2023, 10, 24), date(2023, 10, 24),
'adam', 1, 1),
         File(5, 'main.py', date(2023, 10, 19), date(2023, 10, 19),
'rasulov1337', 1, 2),
         File(6, 'wallpaper.png', date(2023, 10, 20), date(2023, 10, 20),
'root', 1024 * 7, 4)]

filefiledirs = [FileFileDirectory(2, 1),
```

```
                FileFileDirectory(1, 2),
                FileFileDirectory(3, 3),
                FileFileDirectory(4, 1),
                FileFileDirectory(4, 2),
                FileFileDirectory(4, 3),
                FileFileDirectory(5, 1),
                FileFileDirectory(6, 4)]


def main():
    one_to_many = [(f, fd) for f in files for fd in filedirs if f.filedir_id
== fd.id]

    print("Task #D1")
    for i in filter(lambda x: x[0].name.endswith('.py'), one_to_many):
        print(i)

    print("\nTask #D2")

    res = []
    counted_fds = set()
    for _, fd in one_to_many:
        if fd.id not in counted_fds:
            counted_fds.add(fd.id)
        else:
            continue

        file_sizes = [i[0].size for i in one_to_many if i[1] == fd]
        res.append((fd, sum(file_sizes) / len(file_sizes)))

    for i in sorted(res, key=lambda x: x[1]):
        print(i)

    print("\nTask #D3")

    many_to_many = [(fd, f) for ffd in filefiledirs for fd in filedirs for f
in files if
                    f.id == ffd.file_id and fd.id == ffd.file_directory_id]

    dict_ = {i: [] for i in filedirs}
    for fd, f in many_to_many:
        dict_[fd].append(f)

    res_3 = [(i, dict_[i]) for i in dict_]
    for i in filter(lambda x: x[0].path.startswith('/root/PyCharm
Projects/'), res_3):
        print(i)


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

Результат выполнения:

Task #D1

(lab2.py, /root/PyCharm Projects/lab2)

(lab1.py, /root/PyCharm Projects/lab1)

(main.py, /root/PyCharm Projects/lab2)

Task #D2

(/root/PyCharm Projects/lab1, 5.5)

(/root/PyCharm Projects/lab3, 10.0)

(/root/PyCharm Projects/lab2, 512.5)

(/root/Desktop/, 7168.0)

Task #D3

(/root/PyCharm Projects/lab1, [lab1.py, .gitignore, main.py])

(/root/PyCharm Projects/lab2, [lab2.py, .gitignore])

(/root/PyCharm Projects/lab3, [readme.txt, .gitignore])