

```
from operator import itemgetter
```

```
class file:
```

```
    """файл"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
        self.size = size
```

```
        self.catId = catId
```

```
class cat:
```

```
    """каталог"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
class fileCat:
```

```
    """
```

```
    'файлы каталога' для реализации
```

```
    связи многие-ко-многим
```

```
    """
```

```
    def __init__(self, catId, fileId):
```

```
        self.catId = catId
```

```
        self.fileId = fileId
```

```
cats = [cat(1, "D:\программы на python"),
```

```
        cat(2, "D:\отчёты по лабам"),
```

```
        cat(3, "C:\игры")]
```

```
files = [file(1, "Hello world.py", 1, 1),
```

```
        file(2, "лаб 2 отчёт", 100, 2),
```

```
        file(3, "game.py", 3000, 1)
```

```
]
```

```
filesCats = [fileCat(1,1),  
             fileCat(2,2),  
             fileCat(1,3),  
             fileCat(3,3)]
```

```
def main():
```

```
    catsId = [c.id for c in cats]
```

```
    oneToMany = [(f.name, f.size, cats[catsId.index(f.catId)].name) for f in files]
```

```
    filesId = [f.id for f in files]
```

```
    manyToMany = [(files[filesId.index(fc.fileId)].name,  
                  files[filesId.index(fc.fileId)].size,  
                  cats[catsId.index(fc.catId)].name)  
                  for fc in filesCats]
```

```
    print("Task E1")
```

```
    word1 = "r"
```

```
    #выводит названия всех каталогов и всех файлов в них (названия файлов могут повторяться,  
    #отображает пустые каталоги)
```

```
    catsE1 = [c.name for c in cats if word1 in c.name]
```

```
    filesE1 = [otm[0] for otm in oneToMany for c in catsE1 if otm[2] == c]
```

```
    print("каталоги: ", catsE1, "\nфайлы в них: ", filesE1)
```

```
    #выводит полный путь до файла (уникальные записи, не отображает пустые каталоги)
```

```
    #print([otm[2]+"\\ "+otm[0]
```

```
    #    for otm in oneToMany
```

```
    #    if word in otm[2]])
```

```
    print("Task E2")
```

```
    #так и задумывалось
```

```
print(sorted([[c.name, round(sum([otm[1] for otm in oneToMany if otm[2] == c.name)]/(lambda x: 1 if
x==0 else x)(len([otm[1] for otm in oneToMany if otm[2] == c.name]])))] for c in cats],
key=itemgetter(1),reverse=True))
```

```
print("Task E3")
```

```
char3 = "g"
```

```
print([[f.name,[mtm[2] for mtm in manyToMany if mtm[0]==f.name]] for f in files if f.name[0] ==
char3])
```

```
if __name__ == '__main__':
```

```
    main()
```

```
Task E1
каталоги:  ['D:\\программы на python', 'C:\\игры']
файлы в них:  ['Hello world.py', 'game.py']
Task E2
[['D:\\программы на python', 1500], ['D:\\отчёты по лабам', 100], ['C:\\игры', 0]]
Task E3
[['game.py', ['D:\\программы на python', 'C:\\игры']]]
>>>
```