

```
from operator import itemgetter
```

```
class disk:
```

```
    """Диск"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
        self.size = size                #В секундах
```

```
        self.ShelfId = ShelfId
```

```
class Shelf:
```

```
    """Полка"""
```

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

```
        self.id = id
```

```
        self.name = name
```

```
class diskShelf:
```

```
    """
```

```
    'Диски Полки' для реализации
```

```
    связи многие-ко-многим
```

```
    """
```

```
    def __init__(self, ShelfId, diskId):
```

```
        self.ShelfId = ShelfId
```

```
        self.diskId = diskId
```

```
Shelfs = [Shelf(1, "Металл"),
```

```
          Shelf(2, "Рок-Н-Ролл"),
```

```
          Shelf(3, "Хардстайл")]
```

```
disks = [disk(1,"MetallicA",1134,1),
```

```
         disk(2,"Машина времени",2517,2),
```

```
         disk(3,"Bladee Mixtape",1488,3),
```

```
disk(4,"Elvis Presley",1860,2),
disk(5,"SOAD Mezmerize",3252,1)
]
```

```
disksShelfs = [diskShelf(1,1),
               diskShelf(2,2),
               diskShelf(1,3),
               diskShelf(3,3)]
```

```
def main():
```

```
    ShelfsId = [c.id for c in Shelfs]
```

```
    oneToMany = [(f.name, f.size, Shelfs[ShelfsId.index(f.ShelfId)].name) for f in disks]
```

```
    disksId = [f.id for f in disks]
```

```
    manyToMany = [(disks[disksId.index(fc.diskId)].name,
                   disks[disksId.index(fc.diskId)].size,
                   Shelfs[ShelfsId.index(fc.ShelfId)].name)
                  for fc in disksShelfs]
```

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

```
    word1 = "a"
```

```
    #выводит названия всех Полков и всех Дисков в них (названия Дисков могут повторяться,
    #отображает пустые Диски)
```

```
    ShelfsE1 = [c.name for c in Shelfs if word1 in c.name]
```

```
    disksE1 = [otm[0] for otm in oneToMany for c in ShelfsE1 if otm[2] == c]
```

```
    print("Полки: ", ShelfsE1, "\nДиски в них: ", disksE1)
```

```
    #выводит полный путь до Диска (уникальные записи, не отображает пустые Полки)
```

```
    #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 Shelves],  
key=itemgetter(1),reverse=True))
```

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

```
char3 = "M"
```

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

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

```
    main()
```

```
C:\Users\JohnLasley\AppData\Local\Programs\Python\Python312\python.exe "C:\Users\JohnLasley\Desktop\PK-1\PK1 14E.py"  
Task E1  
Полки:  ['Металл', 'Хардстайл']  
Диски в них:  ['Metallica', 'Bladee Mixtape', 'SOAD Mezmerize']  
Task E2  
[['Металл', 2193], ['Рок-Н-Ролл', 2188], ['Хардстайл', 1488]]  
Task E3  
[['Metallica', ['Металл']], ['Машина времени', ['Рок-Н-Ролл']]]  
  
Process finished with exit code 0
```