Текст программы:

```
from operator import itemgetter


class Language:
    def __init__(self, id, name, complexity):
        self.id = id
        self.name = name
        self.complexity = complexity


class Library:
    def __init__(self, id, name, language_id, symbols):
        self.id = id
        self.name = name
        self.language_id = language_id
        self.symbols = symbols


class LanguageLibrary:
    def __init__(self, language_id, lib_id):
        self.language_id = language_id
        self.lib_id = lib_id


languages = [
    Language(1, 'Python', 'medium'),
    Language(2, 'C++', 'medium'),
    Language(3, 'C', 'easy'),
    Language(4, 'Swift', 'hard'),
    Language(5, 'JavaScript', 'medium'),
    Language(6, 'Java', 'medium'),
    Language(7, 'C#', 'medium'),
]


libs = [
    Library(1, 'requests', 2, 100),
    Library(2, 'system', 2, 200),
    Library(3, 'numbers', 6, 250),
    Library(4, 'math', 1, 150),
    Library(5, 'menu', 1, 100),
    Library(6, 'play', 6, 50),
    Library(7, 'reject', 5, 250),
    Library(8, 'formuli', 3, 150),
    Library(9, 'picture', 4, 100),
    Library(10, 'css', 7, 200),
]

language_libs = [
    LanguageLibrary(1, 4),
    LanguageLibrary(1, 5),
    LanguageLibrary(2, 1),
    LanguageLibrary(2, 2),
    LanguageLibrary(3, 8),
    LanguageLibrary(4, 9),
    LanguageLibrary(5, 7),
    LanguageLibrary(6, 6),
    LanguageLibrary(6, 3),
    LanguageLibrary(7, 10),
]

def main():
    one_to_many = [(language.name, language.complexity, lib.name, lib.symbols)
                   for language in languages
                   for lib in libs
                   if lib.language_id == language.id]

    many_to_many_temp = [(language.name, language_lib.language_id, language_lib.lib_id)
                         for language_lib in language_libs
                         for language in languages
                         if language.id == language_lib.language_id]
    many_to_many = [(lib.name, language_name)
                    for language_name, language_id, lib_id in many_to_many_temp
                    for lib in libs if lib.id == lib_id]

    print('Задание E1')
    print(list(filter(lambda i: 'C' in i[0], one_to_many)))

    print('Задание E2')
    language_symbols = {}
    for language in languages:
        language_symbols[language.name] = []
    for row in one_to_many:
        language_name, _, _, symbols = row
        language_symbols[language_name].append(symbols)
    res_12 = [(language, round(sum(symbols) / len(symbols), 2))
              for language, symbols in language_symbols.items() if symbols]
    res_12 = sorted(res_12, key=itemgetter(1), reverse=True)
    print(res_12)

    print('Задание E3')
    res_13 = list(filter(lambda i: i[0][0] == 'm', many_to_many))
    print(res_13)

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

Результат выполнения

```
Задание E1
[('C++', 'medium', 'requests', 100), ('C++', 'medium', 'system', 200), ('C', 'easy', 'formuli', 150), ('C#', 'medium', 'css', 200)]
```

Задание E2

```
[('JavaScript', 250.0), ('C#', 200.0), ('C++', 150.0), ('C', 150.0), ('Java', 150.0), ('Python', 125.0), ('Swift', 100.0)]
```

Задание E3

```
[('math', 'Python'), ('menu', 'Python')]
```