Задание 1

```python
from operator import itemgetter


class Language:
    """Язык программирования"""
    def __init__(self, id, name, popularity, library_id):
        self.id = id
        self.name = name
        self.popularity = popularity
        self.library_id = library_id


class Library:
    """Библиотека"""
    def __init__(self, id, name):
        self.id = id
        self.name = name


class LibraryLanguage:
    """Связь библиотек и языков (многие-ко-многим)"""
    def __init__(self, library_id, language_id):
        self.library_id = library_id
        self.language_id = language_id
```

```python
def one_to_many(libraries, languages):
    """Соединение данных один-ко-многим"""
    return [
        (lang.name, lang.popularity, lib.name)
        for lib in libraries
        for lang in languages
        if lang.library_id == lib.id
    ]


def many_to_many(libraries, libraries_languages, languages):
    """Соединение данных многие-ко-многим"""
    many_to_many_temp = [
        (lib.name, ll.library_id, ll.language_id)
        for lib in libraries
        for ll in libraries_languages
        if lib.id == ll.library_id
    ]
    return [
        (lang.name, lang.popularity, library_name)
        for library_name, library_id, language_id in many_to_many_temp
        for lang in languages
        if lang.id == language_id
    ]
```

```python
def task_g1(libraries, one_to_many_data):
    """Задание Г1"""
    return [
        (lib.name, [lang[0] for lang in one_to_many_data if lang[2] == lib.name])
        for lib in libraries if lib.name.startswith("А")
    ]


def task_g2(libraries, languages):
    """Задание Г2"""
    result = [
        (lib.name, max(lang.popularity for lang in languages if lang.library_id == lib.id))
        for lib in libraries if any(lang.library_id == lib.id for lang in languages)
    ]
    return sorted(result, key=lambda x: x[1], reverse=True)


def task_g3(libraries, many_to_many_data):
    """Задание Г3"""
    return {
        lib.name: [lang[0] for lang in many_to_many_data if lang[2] == lib.name]
        for lib in libraries
    }
```

Задание 2

```python
import unittest
from ref import Language, Library, LibraryLanguage, one_to_many, many_to_many, task_g1, task_g2, task_g3

class TestProgram(unittest.TestCase):

    def setUp(self):
        self.libraries = [
            Library(1, 'Аллергики'),
            Library(2, 'Квадроберы'),
            Library(3, 'Арахнофобы'),
        ]
        self.languages = [
            Language(1, 'Python', 100, 1),
            Language(2, 'Nim', 85, 1),
            Language(3, 'JavaScript', 90, 2),
            Language(4, 'C++', 75, 2),
            Language(5, 'Rust', 80, 3),
            Language(6, 'Assembler', 79, 3),
        ]
        self.libraries_languages = [
            LibraryLanguage(1, 1),
            LibraryLanguage(1, 2),
            LibraryLanguage(2, 3),
            LibraryLanguage(2, 4),
            LibraryLanguage(3, 5),
            LibraryLanguage(3, 6),
        ]

    def test_task_g1(self):
        one_to_many_data = one_to_many(self.libraries, self.languages)
        result = task_g1(self.libraries, one_to_many_data)
        self.assertEqual(result, [
            ('Аллергики', ['Python', 'Nim']),
            ('Арахнофобы', ['Rust', 'Assembler']),
        ])

    def test_task_g2(self):
        result = task_g2(self.libraries, self.languages)
        self.assertEqual(result, [
            ('Аллергики', 100),
            ('Квадроберы', 90),
            ('Арахнофобы', 80),
        ])
```

```python
    def test_task_g3(self):
        many_to_many_data = many_to_many(self.libraries, self.libraries_languages, self.languages)
        result = task_g3(self.libraries, many_to_many_data)
        self.assertEqual(result, {
            'Аллергики': ['Python', 'Nim'],
            'Квадроберы': ['JavaScript', 'C++'],
            'Арахнофобы': ['Rust', 'Assembler'],
        })

if __name__ == "__main__":
    unittest.main()
```

```
...
----------------------------------------------------------------------
Ran 3 tests in 0.001s

OK
```