

## Рубежный контроль 1

Номер варианта:

21	Оператор	Язык программирования
----	----------	--------------------------

Вариант Б:

1. «Язык программирования» и «Оператор» связаны соотношением один-ко-многим. Выведите список всех связанных операторов и языков программирования, отсортированный по операторам, сортировка по языкам программирования произвольная.
2. «Язык программирования» и «Оператор» связаны соотношением один-ко-многим. Выведите список языков программирования с количеством операторов в каждом языке программирования, отсортированный по количеству операторов.
3. «Язык программирования» и «Оператор» связаны соотношением многие-ко-многим. Выведите список все операторы, у которых название заканчивается на «s», и названия их языков программирования.

### Пример текста программы:

```
from operator import itemgetter
```

```
class ProgramLang:
```

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

```
class Operator:
```

```
    def __init__(self, id, name, ascii_code, prog_lang_id):  
        self.id = id  
        self.name = name  
        self.ascii_code = ascii_code  
        self.prog_lang_id = prog_lang_id
```

```
class ProgLangOp:
```

```
    def __init__(self, prog_lang_id, op_id):  
        self.prog_lang_id = prog_lang_id  
        self.op_id = op_id
```

```
prog_langs = [
```

```

    ProgramLang(1, "C++"),
    ProgramLang(2, "Java"),
    ProgramLang(3, "Python"),
]

```

```

ops = [
    Operator(1, "Plus", 43,1),
    Operator(2, "Minus", 45,2),
    Operator(3, "Equal", 61,3),
    Operator(4, "Ampersand", 38, 3),
    Operator(5, "Divide", 47, 1),
    Operator(5, "Multiply", 42, 1)
]

```

```

pl_ops = [
    ProgLangOp(1, 1),
    ProgLangOp(2, 2),
    ProgLangOp(3, 3),
    ProgLangOp(3, 4),
    ProgLangOp(1, 5),
]

```

```

def main():
    one_to_many = [(op.name, op.ascii_code, pl.name)
                    for pl in prog_langs
                    for op in ops
                    if op.prog_lang_id == pl.id]

    many_to_many_temp = [(pl.name, ps.prog_lang_id, ps.op_id)
                          for pl in prog_langs
                          for ps in pl_ops
                          if ps.prog_lang_id == pl.id]

    many_to_many = [(op.name, op.ascii_code, pl_name)
                    for pl_name, pl_id, op_id in many_to_many_temp
                    for op in ops if op.id == op_id]

    print("Задание Б1")
    res_1 = sorted(one_to_many, key=itemgetter(0))
    print(res_1)

    print("\nЗадание Б2")
    res_2 = []
    temp_dict = dict()
    for i in one_to_many:
        if i[2] in temp_dict:
            temp_dict[i[2]] += 1
        else:
            temp_dict[i[2]] = 1
    for i in temp_dict.keys():
        res_2.append((i, temp_dict[i]))

    res_2.sort(key=itemgetter(1), reverse=True)
    print(res_2)

```

```
print("\nЗадание Б3")
res_3 = [(i[0], i[2]) for i in many_to_many if i[0].endswith('s')]
print(res_3)
```

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

### **Результат выполнения:**

Задание Б1

```
[('Ampersand', 38, 'Python'), ('Divide', 47, 'C++'), ('Equal', 61, 'Python'), ('Minus', 45, 'Java'), ('Multiply', 42, 'C++'), ('Plus', 43, 'C++')]
```

Задание Б2

```
[('C++', 3), ('Python', 2), ('Java', 1)]
```

Задание Б3

```
[('Plus', 'C++'), ('Minus', 'Java')]
```