Текст программы

```
# 1. Street -> House 1-to-many. Print all houses' owners living in streets, sorted by streets.
# 1. Street -> House 1-to-many. Вывести всех владельцев домов по улицам, отсортировать по улицам.

# 2. Street -> House 1-to-many. Print all the streets with the total cost of all houses at the street, sorted by total cost.
# 2. Street -> House 1-to-many. Вывести все улицы и суммарную стоимость расположенных на ней домов, отсортировать по суммарной стоимости.

# 3. Street -> House many-to-many. Print all the streets which contain diin their name and all the owners living on these streets.
# 3. Street -> House many-to-many. Вывести всех владельцев домов только на тех улицах, в названии которых есть цифры.

from operator import attrgetter
from operator import itemgetter


class Street:
    def __init__(self, street_name) -> None:
        self.name = street_name


class House:
    def __init__(self, owner, price, number, street) -> None:
        self.owner = owner
        self.price = price
        self.number = number
        self.street = street


class HousesToStreet:
    def __init__(self, street, number) -> None:
        self.street = street
        self.number = number


streets: list = [Street("Lenina"), Street("Yunosti"),
                 Street("Vertolyotnaya"), Street("8 Marta"),
                 Street("40 Let Pobedi"), Street("1905 goda")]


houses: list = [House("Ivanov", 10000, 1, "Yunosti"),
```

```
                House("Petrov", 5000, 1, "Lenina"),
                House("Sergeev", 17000, 4, "8 Marta"),
                House("Malishev", 5000, 3, "8 Marta"),
                House("Povarov", 45000, 1, "40 Let Pobedi"),
                House("Govorov", 35000, 2, "40 Let Pobedi"),
                House("Voronov", 15000, 3, "40 Let Pobedi"),
                House("Sidorov", 42000, 2, "Yunosti"),
                House("Musk", 35000, 1, "8 Marta"),
                House("Krivov", 35000, 1, "1905 goda"),
                House("Torvalds", 12345, 2, "8 Marta")]


houses_to_streets: list = [HousesToStreet("Lenina", 1),
                           HousesToStreet("8 Marta", 1),
                           HousesToStreet("8 Marta", 2),
                           HousesToStreet("Yunosti", 1),
                           HousesToStreet("Yunosti", 2),
                           HousesToStreet("8 Marta", 3),
                           HousesToStreet("8 Marta", 4),
                           HousesToStreet("40 Let Pobedi", 1),
                           HousesToStreet("40 Let Pobedi", 2),
                           HousesToStreet("40 Let Pobedi", 3),
                           HousesToStreet("1905 goda", 1),
                           ]


def test1():
    one_to_many: list = [(h.street, h.owner)
                         for s in sorted(streets, key=attrgetter("name"))
                         for h in houses
                         if h.street == s.name]

    for street_name, owner_name in one_to_many:
        print(f"{street_name}: {owner_name}")


def test2():
    total_costs: list = sorted([(street.name, sum([h.price for h in houses if h.street
== street.name]))
                                for street in sorted(streets, key=attrgetter("name"))],
key=itemgetter(1))

    for street, total_cost in total_costs:
        print(f'{street} -> {total_cost}')
```

```python
def is_contains_number(input: str) -> bool:
    return any(ch.isdigit() for ch in input)


def test3():

    many_to_many = [(hts.street, h.owner)
                    for hts in sorted(houses_to_streets, key=attrgetter("street")) if
is_contains_number(hts.street)
                    for h in houses if (h.number == hts.number and h.street ==
hts.street)]

    ans = {s: [h for inner_s, h in many_to_many if inner_s == s]
           for s, _ in many_to_many}

    print(ans)



if __name__ == "__main__":

    print("Task 1:")
    test1()
    print("\n\n")

    print("Task 2:")
    test2()
    print("\n\n")

    print("Task 3:")
    test3()
    print("\n\n")
```

Результат тестирования:

Task 1:
1905 goda: Krivov
40 Let Pobedi: Povarov
40 Let Pobedi: Govorov
40 Let Pobedi: Voronov
8 Marta: Sergeev
8 Marta: Malishev
8 Marta: Musk
8 Marta: Torvalds
Lenina: Petrov

Yunosti: Ivanov
Yunosti: Sidorov

Task 2:
Vertolyotnaya -> 0
Lenina -> 5000
1905 goda -> 35000
Yunosti -> 52000
8 Marta -> 69345
40 Let Pobedi -> 95000

Task 3:
{'1905 goda': ['Krivov'], '40 Let Pobedi': ['Povarov', 'Govorov', 'Voronov'], '8 Marta': ['Musk', 'Torvalds', 'Malishev', 'Sergeev']}