

```

from random import randint
# используется для сортировки
from operator import itemgetter

class Composition:
    '''Класс Музыкальная Композиция

    ...

    def __init__(self,id,name,duration,comp_id):
        self.id=id
        self.name=name
        self.duration=duration
        self.comp_id=comp_id
class Orchestra:

    def __init__(self,id,name) -> None:
        '''Принимает уникальный индекс оркестра,
        его название, кол-во сыгранных композиций
        Оркестр может сыграть
        ...

        self.id=id
        self.name=name

class Orchestra_to_Composition:
    '''Обеспечивает связь М:М
    Между классами Оркестр и Композиция
    ...

    def __init__(self,Orchestra_id,Composition_id):
        self.orc_id=Orchestra_id
        self.comp_id=Composition_id

orchestras=[
    Orchestra(1,"Лондонский симфонический оркестр"),
    Orchestra(2,"Московский оркестр"),
    Orchestra(3,"Белорусский классический оркестр"),
    Orchestra(4,"Токийский новый оркестр")
]

composition=[
    Composition(1,'Undertale','6:20',4),
    Composition(2,'Seasons. Vivaldi','8:00',1),
    Composition(3,'Asgore','4:10',4),
    Composition(4,'Hopes and dreams','3:20',4),
    Composition(5,'Щелкунчик','90:03',2),
    Composition(6,'Гимн Белоруси','2:13',3)
]

```

```

orc_to_comp=[Orchestra_to_Composition(1,1),Orchestra_to_Composition(1,2),
Orchestra_to_Composition(3,3),Orchestra_to_Composition(3,6),Orchestra_to_Composition
(2,2)
]
def main():
    one_to_many=[(c.name,c.duration,o.name)
                  for o in orcestras
                  for c in composition

                  if o.id==c.comp_id]
    print("B1. Сортировка значений по названию в алфавитном порядке")
    print(*sorted(one_to_many,key=itemgetter(0)),sep='\n')

    print("\n\nB2. Сортировка значений по минимальной длительности,
отсортированный по возрастанию")

    print(*sorted(one_to_many,key=itemgetter(1)),sep='\n')

    print("\n\nB3.")
    many_to_many_temp = [(o.name, o_to_c.comp_id, o_to_c.orc_id)
                          for o in orcestras
                          for o_to_c in orc_to_comp
                          if o.id==o_to_c.comp_id]

    many_to_many = [(c.name, c.duration, dep_name)
                    for dep_name, dep_id, emp_id in many_to_many_temp
                    for c in composition if c.id==emp_id]
    print(*sorted(many_to_many,key=itemgetter(0)))

if __name__=='__main__':
    main()

```

B1. Сортировка значений по названию в алфавитном порядке

('Asgore', '4:10', 'Токийский новый оркестр')

('Hopes and dreams', '3:20', 'Токийский новый оркестр')

('Seasons. Vivaldi', '8:00', 'Лондонский симфонический оркестр')

('Undertale', '6:20', 'Токийский новый оркестр')

('Гимн Белоруси', '2:13', 'Белорусский классический оркестр')

('Щелкунчик', '90:03', 'Московский оркестр')

В2. Сортировка значений по минимальной длительности, отсортированный по возрастанию

('Гимн Белоруси', '2:13', 'Белорусский классический оркестр')

('Hopes and dreams', '3:20', 'Токийский новый оркестр')

('Asgore', '4:10', 'Токийский новый оркестр')

('Undertale', '6:20', 'Токийский новый оркестр')

('Seasons. Vivaldi', '8:00', 'Лондонский симфонический оркестр')

('Щелкунчик', '90:03', 'Московский оркестр')

В3.

('Asgore', '4:10', 'Белорусский классический оркестр') ('Seasons. Vivaldi', '8:00', 'Московский оркестр') ('Undertale', '6:20', 'Лондонский симфонический оркестр') ('Undertale', '6:20', 'Московский оркестр')