

Задача 3. Файлы

Источник:	базовая
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Вася хочет найти кое-какой интересный файл в списке, но в сожалению не помнит его имени. Он только помнит, что создал этот файл не раньше, чем в момент времени A , и что после момента времени B он точно его **не** менял. Кроме того, он абсолютно уверен, что этот файл был виден при просмотре. Нужно написать программу, которая будет отбирать файлы, подходящие под этот критерий, и выдавать их Васе.

Список файлов и директорий задан в бинарном виде. Все целые числа записаны с little-endian порядком байтов. Любой момент времени (timestamp) задаётся как количество 100-наносекундных интервалов, прошедших с 1 января 1601 года (текущий момент времени примерно равен 131 832 294 671 670 965). Далее и файлы, и директории мы будем называть обобщённо «файлами».

Входные данные заданы в следующем формате (слева указан диапазон байтов):

0-3	целое число N — количество файлов в списке ($1 \leq N \leq 1\,000$)
4-11	целое число A — искомый файл был создан не раньше этого момента времени ($0 < A < 10^{18}$)
12-19	целое число B — искомый файл не модифицировался позже этого момента времени ($A < B < 10^{18}$)
20-??	В оставшейся части входных данных подряд идут описания N файлов.

Каждый отдельный файл описывается непрерывным блоком данных в формате:

0-20	Имя файла, состоящее из латинских букв, цифр и символов точки. Длина имени не превышает 20, все оставшиеся байты заполнены нулями.
21-28	Размер S_i файла в байтах ($0 \leq S_i \leq 10^{12}$).
29	Байт равен 1, если файл на самом деле является директорией, и 0 иначе.
30-37	Момент времени C_i , когда файл был создан ($0 < C_i < 10^{18}$).
38-45	Момент времени M_i , когда файл был изменён в последний раз ($C_i < M_i < 10^{18}$).
46	Байт равен 1, если файл скрытый, и 0 иначе.

Гарантируется, что все заданные файлы имеют разные имена. Нужно выбрать среди них файлы, удовлетворяющие критериям:

1. Не директория и не скрытый.
2. Создан не раньше времени A , последний раз изменён не позже времени B .

Эти файлы нужно отсортировать по имени в лексикографическом порядке, после чего вывести их все подряд в выходные данные. Выводить каждый файл следует блоком данных в том же формате, в котором он был задан во входных данных. Учтите, что при лексикографическом сравнении символы сравниваются по их ASCII-коду.

Замечание: Чтобы не писать много кода, можно читать и писать файл целиком как структуру.

Пример

Пример входных и выходных данных можно скачать по ссылке.