

# Финал - Бэкенд-разработка

 1 июн 2019, 16:00:28

старт: 1 июн 2019, 12:00:00

финиш: 1 июн 2019, 16:00:00

длительность: 04:00:00

начало: 1 июн 2019, 12:00:00

конец: 1 июн 2019, 16:00:00

## К. Список писем

Ограничение времени	3 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Электронное письмо можно упрощённо представить в виде структуры

```
struct Message {  
    int userId; // id получателя  
    string from; // email-адрес отправителя  
    unsigned receiveTimestamp; // время получения  
    int folderId; // id папки, в которую доставлено письмо  
    int threadId; // id цепочки (треда), в которую группируются письма при обсуждении  
};
```

Нужно реализовать хранилище писем, в которое можно складывать письма и у которого можно запрашивать список писем по параметрам. Параметрами для запроса являются:

1. id юзера
2. id папки
3. отрезок времени получения письма , включительно
4. максимальное количество элементов в выдаче

Письма в выдаче должны быть сгруппированы в треда (сначала идут письма первого треда, затем второго и так далее). Таймстемпом треда считается таймстемп самого свежего письма в этом треде.

Должны выбираться только треда, принадлежащие папке . Считается, что тред принадлежит папке , если хотя бы одно письмо из этого треда лежит в папке . Таким образом в выдаче писем могут присутствовать письма с , отличающимся от переданного.

В выдачу должны попасть только треда, таймстемпы которых попадают в отрезок , включительно.

Треда в выдаче должны быть отсортированы по убыванию своих таймстемпов. Для тредов с одинаковым таймстемпом сортировка идет по увеличению . Письма внутри треда должны быть отсортированы по убыванию таймстемпа. Для писем с одинаковым таймстемпом сортировка идет по порядку сохранения в хранилище.

После всех группировок применяется правило для ограничения количества элементов в выдаче: надо выбрать первых.

## Формат ввода

На вход программе подается последовательность строк вида

```
n q  
+ <userId> <from> <timestamp> <folderId> <threadId>  
...  
? <userId> <folderId> <since> <till> <count>  
...
```

1. Первая строка содержит два целых числа (количество запросов на добавление письма, ) и (кол-во запросов на выборку писем, )
2. после первой строки идет запросов: (действие "сложить письмо") или (действие "получить выдачу"), действия могут чередоваться
3. , , - произвольные целые числа, , ,
4. , , - произвольные целые числа в отрезке , причем
- 5.
6. - строка из символов , длиной , где

## Формат вывода

Программа должна для каждой операции выдать следующую секцию:

М

<userId> <from> <timestamp> <folderId> <threadId>

...

<userId> <from> <timestamp> <folderId> <threadId>

где - количество элементов в выдаче на запрос . После строки с числом следует столько же строк, каждое обозначает письмо. Количество таких секций равняется количеству действий во входном файле.

### Пример 1

Ввод	Вывод
1 1 + 1 hello@yandex.ru 5 1 1 ? 1 1 0 10 10	1 1 hello@yandex.ru 5 1 1

### Пример 2

Ввод	Вывод
2 1 + 1 hello1@yandex.ru 1 1 1 + 1 hello2@yandex.ru 2 1 2 ? 1 1 0 10 1	1 1 hello2@yandex.ru 2 1 2

Язык Моно C# 5.2.0

Набрать здесь

Отправить файл

1

Отправить

Предыдущая

Следующая

