

Школа бэкенд-разработки 2021 (осень)

🕒 7 сен 2021, 22:49:42

старт: 7 сен 2021, 19:54:21

финиш: 8 сен 2021, 01:54:21

до финиша: 03:04:24

длительность: 06:00:00

С. Мониторинг логов

Ограничение времени	3 секунды
Ограничение памяти	512.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В команде разработки бэкенда одной известной компании задумались об автоматическом информировании об ошибках в системе. Для этого необходимо реализовать программу, анализирующую логи выбранной машины и определяющую первый момент времени, ставший критическим.

Критическим называется такой момент времени K , что на промежутке $[K - t + 1; K]$ произошло суммарно ошибок больше либо равно e .

Ошибками являются все сообщения, имеющие статус **ERROR**.

Формат ввода

В первой строке записаны два целых числа t и e ($1 \leq t \leq 86400$; $1 \leq e \leq 200$) - размер промежутка времени для вычисления критического момента и минимально необходимое суммарное число ошибок на промежутке.

Последующие строки описывают лог машины. Общее количество строк не превышает $3 \cdot 10^5$. Каждая строка задаёт ровно одно сообщение в формате

```
[yyyy-MM-dd hh:mm:ss] STATUS MESSAGE
```

Дата и время сообщения принимают значения от 2020-01-01 00:00:00 до 2021-12-31 23:59:59 и находятся во временной зоне **UTC(+0)**. Сообщения отсортированы в **хронологическом порядке**. Напоминаем, что 2020 - **високосный год**.

Поле STATUS может принимать значения



- INFO
- WARNING
- ERROR

Поле MESSAGE может быть любой последовательностью символов ASCII, за исключением разделителей строк, длины меньше 30 символов.


Формат вывода

Если не существует критического момента времени - выведите -1. Иначе выведите строку в формате yyyy-MM-dd hh:mm:ss — первый критический момент времени, когда количество ошибок за последние t секунд стало не менее e .

Пример 1

Ввод Вывод 

Ввод 


Вывод 

```
60 3
[2020-03-16 16:15:25] INFO Disk size is 100 Gb
[2020-03-16 16:15:25] ERROR Db failute
[2020-03-16 16:15:25] ERROR Network failute
[2020-03-16 16:16:29] ERROR Cant write varlog
[2020-03-16 16:16:42] ERROR Unable to start process
[2020-03-16 16:16:43] WARNING Disk size is too small
[2020-03-16 16:16:43] ERROR Config not found
[2020-03-16 16:16:53] ERROR Timeout detected
```

2020-03-16 16:16:43

Пример 2

Ввод 


Вывод 

```
1 2
[2020-03-16 23:59:59] ERROR Disk crash
[2020-03-17 00:00:00] ERROR Network failute
[2020-03-17 00:00:01] ERROR Cant write varlog
```

-1

Пример 3

Ввод 

Вывод 

```
2 2
[2020-03-16 16:15:25] INFO Disk size is 100 Gb
[2020-03-16 23:59:59] ERROR Disk size is too sm
[2020-03-17 00:00:00] ERROR Network failute dete
[2020-03-17 00:00:01] ERROR Cant write varlogmysq
```

2020-03-17 00:00:00

Примечания

В примере под номером 1 выбираются сообщения
[2020-03-16 16:16:29] ERROR Cant write varlog
[2020-03-16 16:16:42] ERROR Unable to start process
[2020-03-16 16:16:43] ERROR Config not found

В примере под номером 2 нет подходящих сообщений. Чтобы ответ был не -1, достаточно чтобы любые два сообщения пришли в одну и ту же секунду.

В примере под номером 3 выбираются сообщения
[2020-03-16 23:59:59] ERROR Disk size is too sm
[2020-03-17 00:00:00] ERROR Network failute dete

Язык OpenJDK Java 15 

Набрать здесь

Отправить файл

```

1 import java.io.FileNotFoundException;
2 import java.io.FileReader;
3 import java.text.ParseException;
4 import java.text.SimpleDateFormat;
5 import java.util.*;
6
7 public class C {
8     public static void main(String[] args) throws FileNotFoundException, ParseException {
9         FileReader reader = new FileReader("input.txt");
10        //FileReader reader = new FileReader("input.txt");
11        Scanner scanner = new Scanner(reader);
12        int t = scanner.nextInt();
13        int e = scanner.nextInt();
14        List<String> str = new LinkedList<>();
15
16        while(scanner.hasNext()){
17            str.add(scanner.nextLine());
18        }
19        List<Integer> time = new LinkedList<>();
20        List<Integer> timegood = new LinkedList<>();
21        timegood.add(0);
22
23        List<String> datesTime = new LinkedList<>();
24        List<String[]> list = new LinkedList<>();
25        for(int i=1; i<str.size();i++){
26            //System.out.println(str.get(i));
27            String string = str.get(i);
28            String[] words = string.split(" ");
29            //System.out.println(Arrays.toString(words));
30            list.add(words);
31
32            String forDatesTime = words[0].substring(1,words[0].length())+" "+ words[1].substring(0,words[1].length()-1);
33
34            if(words[2].equals("ERROR")){
35                //System.out.println(words[2]);
36                datesTime.add(words[0].substring(1,words[0].length())+" "+ words[1].substring(0,words[1].length()-1));
37                if(Integer.parseInt(forDatesTime.substring(11,13))==0){
38

```

Отправить

Предыдущая

Следующая