

Тестовое задание:

Имеется access-лог web-сервера. Файл со следующей структурой.

```
192.168.32.181 - - [14/06/2017:16:47:02 +1000] "PUT /rest/v1.4/documents?zone=default&_rid=6076537c
HTTP/1.1" 200 2 44.510983 "-" "@list-item-updater" prio:0
```

```
192.168.32.181 - - [14/06/2017:16:47:02 +1000] "PUT /rest/v1.4/documents?zone=default&_rid=7ae28555
HTTP/1.1" 200 2 23.251219 "-" "@list-item-updater" prio:0
```

```
192.168.32.181 - - [14/06/2017:16:47:02 +1000] "PUT /rest/v1.4/documents?zone=default&_rid=e356713
HTTP/1.1" 200 2 30.164372 "-" "@list-item-updater" prio:0
```

У каждой записи есть HTTP-код ответа (9-е поле, в первом примере "200") и время обработки запроса в миллисекундах (11-е поле, в первом примере: "44.510983"). Каждый день оператор выполняет анализ лога локализуя диапазоны времени, когда доля отказов сервиса превышала указанную границу. С этими инцидентами позже разбирается группа разработки. Требуется написать алгоритм читающий access-лог и выполняющий анализ отказов автоматически.

Отказом считается запрос завершившийся с любым 500-м кодом возврата (5xx) или обрабатываемый дольше чем указанный интервал времени.

На входе программе дается:

- * поток данных из access-лог'a;
- * минимально допустимый уровень доступности (проценты. Например, "99.9");
- * приемлемое время ответа (миллисекунды. Например, "45").

На выходе программа предоставляет временные интервалы, в которые доля отказов системы превышала указанную границу, а также уровень доступности в этот интервал времени. Интервалы должны быть отсортированы по времени начала.

Пример использования программы:

```
$ cat access.log | php analyze.php -u 99.9 -t 45
```

```
13:32:26 13:33:15 94.5
```

```
15:23:02 15:23:08 99.8
```

Требования и ограничения.

- * максимальный размер access-лог'a не позволяет загрузить все записи в оперативную память. Анализ необходимо выполнять потоково. Объем доступной памяти 512 мегабайт (memory_limit 512M);
- * допускается использования версии PHP 7.4;
- * допускается использование сторонних библиотек;
- * в качестве менеджера зависимостей должен использоваться composer;
- * проект должен содержать автоматические тесты фиксирующие поведение системы в объеме по вашему усмотрению;
- * инструкция по установке и запуску должна находиться в readme.md в корне проекта;
- * приветствуется наличие dockerfile разворачивающего приложение в контейнере;