

## ip\_filter

### Условие

Программа из стандартного ввода читает данные. Данные хранятся построчно. Каждая строка состоит из трех полей разделенных одним символом табуляции и завершается символом конца строки.

```
text1 \t text2 \t text3 \n
```

Поля *text2* и *text3* игнорируются. Поле *text1* имеет следующую структуру (ip4 address):

```
n1.n2.n3.n4
```

где *n1..4* - число от 0 до 255

Необходимо загрузить список ip-адресов в память и вывести в следующем порядке:

Список адресов отсортированных в обратном лексикографическом порядке адресов. Одна строка один адрес.

Пример лексикографической сортировки (по первому числу, затем по второму и так далее):

```
1.1.1.1
1.2.1.1
1.10.1.1
```

Соответственно обратная:

```
1.10.1.1
1.2.1.1
1.1.1.1
```

Сразу продолжается список адресов первый байт которых равен 1. Порядок сортировки не меняется. Одна строка один адрес. Списки ничем не разделяются.

Сразу продолжается список адресов первый байт которых равен 46, а второй 70. Порядок сортировки не меняется. Одна строка один адрес. Списки ничем не разделяются.

Сразу продолжается список адресов любой байт которых равен 46. Порядок сортировки не меняется. Одна строка один адрес. Списки ничем не разделяются.

## Требования к реализации

В приложенном к заданию исходном файле необходимо заменить, где это возможно конструкции на аналогичные из стандарта C++14.

Реализовать недостающий функционал.

Выдержка из файла со списком адресов к задаче прилагается.

## Самоконтроль

- не пришлось создавать новый Package в bintray
- смогли решить задачу сортировки так и не написав ни одной лямбды
- не стали приводить ip-адрес к типу `int/long`
- параметры фильтров не закодированы в алгоритм фильтрации
- пакет `ip_filter` содержащий исполняемый файл `ip_filter` опубликован на bintray
- отправлена на проверку ссылка на страницу репозитория

## Проверка

Задание считается выполнено успешно, если после просмотра кода, подключения репозитория, установки пакета и запуска бинарного файла командой:

```
$ cat ip.tsv | ip_filter > out.tsv
```

При эталонном `ip.tsv` полученный `out.tsv` совпадет с эталонным.

Будет отмечена скорость выполнения задания. Наличие тестов. Покрытие возможностями C++14.