

Задание 4

Условие

Реализовать функцию печати условного ip-адреса. Требуется рассмотреть несколько случаев.

Адрес может быть представлен в виде произвольного целочисленного типа. Представление не зависит от знака типа. Выводить побайтово начиная со старшего с символом `.` в качестве разделителя.

Пример:

для входного значения `int{1234}` результат будет `"0.0.4.210"` (при `sizeof(int) == 4`)

Адрес может быть представлен в виде строки. Выводится как есть.

Пример:

для входного значения `std::string{"trololo"}` результат будет `"trololo"`

Адрес может быть представлен в виде контейнеров `std::list`, `std::vector`. Выводится содержимое контейнера поэлементно и разделяется символом `.` (одним символом точка).

Пример:

для входного значения `std::vector{0, 1, 2, 3}` результат будет `"0.1.2.3"`

Опционально адрес может быть представлен в виде `std::tuple` при условии, что все типы одинаковы. Выводится содержимое поэлементно и разделяется `.` (одним символом точка).

Пример:

для входного значения `std::make_tuple(0, 1, 2, 3)` результат будет `"0.1.2.3"`

Прикладной код должен содержать следующие вызовы:

- Печать адреса как `char(-1)`
- Печать адреса как `short(0)`
- Печать адреса как `int(2130706433)`
- Печать адреса как `long(8875824491850138409)`
- Печать адреса как `std::string`
- Печать адреса как `std::vector`
- Печать адреса как `std::list`
- Опционально. Печать адреса как `std::tuple`

Также требуется добавить в `.travis.yml` на этапе сборки вызов `doxygen` и публикацию html-версии документации на `github-pages`. Подробное описание на странице:

<https://docs.travis-ci.com/user/deployment/pages/>

Включить в репозиторий файл `Doxyfile` с включенными опциями `HAVE_DOT` и `EXTRACT_ALL`.

Самоконтроль

- функция печати должна быть одной шаблонной функцией с различными вариантами реализации через механизм SFINAR
- вариант реализации для целочисленного представления должен быть единственным
- вариант реализации для контейнеров должен быть единственным
- не должно быть реализации типа "если не совпало с иными - значит это контейнер"
- опциональная реализация для `std::tuple` должна приводить к ошибке в случае отличающихся типов
- найдите самый простой способ сделать вариант для печати `std::string`
- бинарный файл и пакет `print_ip` содержащий бинарный файл `print_ip` опубликован на `bintray`

Проверка

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

```
print_ip
```

будут выведены адреса

```
255
```

```
0.0
```

```
127.0.0.1
```

```
123.45.67.89.101.112.131.41
```

вслед за которыми будут выведены адреса из строки, контейнеров и опционально из кортежа.