

nickname

Условие

Пользуясь имеющимися в библиотеке STL структурами и алгоритмами разработать новую структуру Radix Tree и операции над ней.

Пользуясь полученной структурой определить минимальное количество символов достаточных для однозначного определения пользователя по его имени, предварительного добавленных в новый контейнер из стандартного ввода.

Например, для имен Данил, Денис и Даша мы хотим получить

Данил - Дан

Денис - Де

Даша - Даш

Radix Tree

Пример структуры узла:

```
class RadixTrie {
    struct Node {
        std::string label;
        bool is_end{};
        std::array<std::unique_ptr<Node>, 26> children;
    };
    std::unique_ptr<Node> root;
public:
    static void append(std::unique_ptr<Node> &node, const std::string &label) {
        // TODO
    }

    void append(const std::string &label) {
        append(root, label);
    }
};
```

Ссылки для понимания работы структуры:

- [Визуализация RadixTree](#)
- https://en.wikipedia.org/wiki/Radix_tree

Пример

Символ \$ обозначает признак завершения строки. Текст выводится в кавычках для обозначения границ текста.

Добавление "aleksey".

"aleksey"\$

Добавление "sasha". Корень становится пустой строкой без признака завершения.

""

| "aleksey"\$
└ "sasha"\$

Добавление "aleks"

""

| "aleks"\$
| └ "ey"\$
└ "sasha"\$

Добавление "alek"

```
""
└ "alek"$
|  └ "s"$
|    └ "ey"$
└ "sasha"$
```

Добавление "alesha" приводит к расщеплению узла "alek" при этом узел "ale" без признака завершения.

```
""
└ "ale"
|  └ "k"$
|    └ "s"$
|      └ "ey"$
└ "sha"$
└ "sasha"$
```

Добавление "maksim"

```
""
└ "ale"
|  └ "k"$
|    └ "s"$
|      └ "ey"$
└ "sha"$
└ "maksim"$
└ "sasha"$
```

Требуется вывести все элементы контейнера. Список должен получиться идентичный добавленным элементам за исключением порядка. Рядом с каждым элементом вывести минимальный префикс для однозначной идентификации, например:

```
alek alek
aleks aleks
aleksey alekse
alesha ales
maksim m
sasha s
```

Требуется вывести все элементы в виде дерева, уровни обозначать отступами, например

```
ale
  k$
    s$
      ey$
    sha$
maksim$
sasha$
```

Результат

- структура не должна содержать сырых указателей
- пакет nickname содержащий исполняемый файл nickname 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 logins.txt | nickname
```

Будет выведено содержимое контейнера с сокращениями и структура контейнера в виде дерева.

Оценка

максимум 10

баллы	за что
5	Успешно прошла проверка.
2	Дана оценка сложности добавления строки.
1	Связи между узлами отображены специальными символами для рисования рамок.
1	Структура корректно работает с любой однобайтовой кодировкой.
1	Структура корректно работает с UTF-8.