Задача 5

Исследование блогов

Крупная IT-компания Янгл проводит исследование упоминаемости своего бренда в блогах и социальных сетях. Аналитики компании уже выгрузили интересующие их публикации и сформировали набор ключевых слов. Их интересует, сколько раз каждое ключевое слово входит в набор публикаций.

Помогите им это сделать — напишите функцию

Stats ExploreKeyWords(const set<string>& key_words, istream& input).

Её параметры:

- key_words — множество ключевых слов
- input — входной поток, содержащий исследуемые публикации в блогах и соц. сетях, одна строка — одна публикация.

Функция ExploreKeyWords должна возвращать структуру Stats, в которой хранится, сколько раз каждое слово из key_words суммарно встретилось в потоке input:

```
1  struct Stats {
2  map<string, int> word_frequences;
3
4  void operator += (const Stats& other);
5  };
```

При подсчёте вхождения ключевых слов в текст нужно учитывать только вхождения в виде отдельных слов. Слова друг от друга отделяются одним или несколькими пробелами. В качестве примера допустим, что key_words = {"yangle", "rocks", "sucks", "all"} и у нас есть публикация из соц. сети Switter: "Goondex really sucks, but yangle rocks ! Use yangle". Слово «yangle» входит в неё 2 раза, «rocks» — один раз, а слова «all» и «sucks» — ни разу. Слово «sucks» не входит ни разу, потому что в данном примере оно является префиксом слова «sucks,» (по условию, слова друг от друга отделяются только пробелами). Несмотря на то, что слово «all» является частью слова «really», его вхождение тоже нельзя засчитывать, так как это не отдельное слово.

Чтобы уложиться в Time Limit, подумайте, можно ли эту задачу распараллелить.

Замечание

До этого в методических материалах мы показывали, как использовать функцию async для асинхронного запуска лямбда-функций без параметров. Однако её можно использовать и для запуска функций, у которых есть параметры. Например,

```
1  string Join(string s, string t) {
2  return s + t;
3  }
4
5  string a = "Hello";
6  string b = " world";
7  future<string> f = async(Join, a, b);
```

Как видите, чтобы передать параметры в функцию, запускаемую асинхронно, их надо указать в качестве параметров функции async после самой функции. Важно отметить, что при таком вызове будут **созданы копии фактических параметров** (т.е. в нашем примере выше строки a и b будут скопированы). Это не всегда желательно. Например, если у нас есть константный объект, который потоки

только читают, у нас нет необходимости создавать его копию, и можно передать его по ссылке. Для этого надо воспользоваться функцией ref из заголовочного файла <functional>.

```
Stats ExploreKeyWordsSingleThread(const set<string>& key_words, istream& input);

Stats ExploreKeyWords(const set<string>& key_words, istream& input) {
// key_words и input будут переданы по ссылке, а не скопированы
return async(ExploreKeyWordsSingleThread, ref(key_words), ref(input)).get();
}
```

Файл с заготовкой задачи

explore_key_words.cpp