

# Волга ИТ - 2015. Прикладное Программирование

## Очный тур, первый этап

### Введение

Так! Прежде чем рассылать спам, нужно подготовиться. Нам потребуется каким-то образом обходить почтовые фильтры. Хм... Соберем побольше текстов и посмотрим какие они.



### Постановка задачи

Требуется создать приложение, которое загружает из БД SQLite тексты и анализирует их согласно заданию:

#### Частотный словарь (100 баллов):

Необходимо найти 100 самых употребляемых слов во всех текстах базы данных. Если некоторые слова встречаются в равной степени, то они должны быть выведены все.

Слово — комбинация символов на русском и английском языках без знаков препинания и учета регистра.

#### Графический интерфейс (100 баллов):

Приложение должно иметь графический интерфейс и отвечать следующим требованиям:

- Иметь список названий всех текстов из базы данных.
- Возможность просмотреть текст для выбранного из списка названия.
- Возможность посмотреть частотный словарь (слово и частоту его использования).

#### Пример ввода:

Show me your flowchart and conceal your tables, and I shall continue to be mystified. Show me your tables, and I won't usually need your flowchart; it'll be obvious.

#### Пример вывода:

your	4
and	3
show	2
Остальные слова...	

#### Структура базы данных:

База данных SQLite содержит следующие таблицы:

- Таблица «literature» со следующими столбцами:
  - id — уникальный числовой идентификатор текста (primary key)
  - name — название текста
  - content — текст (длина текста не превышает  $8 * 1024 * 1024$  символов)
- Таблица «dictionary»:
  - id — произвольный уникальный числовой идентификатор текста (primary key)
  - word — слово
  - count — как часто данное слово встречается в текстах

Таблица «literature» изначально заполнена и ее нет необходимости изменять. Таблица «dictionary» изначально пуста и должна быть заполнена программой.

## **Дополнительные условия**

Программа предоставляется в виде исполняемого файла, исходных текстов, файлов проекта для среды разработки и readme.txt.

Рекомендуемые среды разработки:

- C++: Qt Creator
- C# : Visual Studio Express
- Java: Eclipse или IntelliJ IDEA

Постарайтесь свести к минимуму все внешние зависимости и использовать только инструменты, предоставляемые Java, .NET и Qt.

В файл `readme.txt` включите

- Краткое описание алгоритма
- Инструкцию по сборке программы из исходных текстов (дополнительные условия, настройки среды - постарайтесь свести их к минимуму)
- Требования для запуска исполняемого файла, если таковые имеются (если приложение не запустится, мы попробуем собрать его заново из исходных текстов, но это минус)

Принимаются частичные решения: GUI реализован частично или отсутствует, вместо БД данные читаются и выводятся в текстовый файл, алгоритм реализован не полностью и пр. В таком случае в файле `readme.txt` необходимо указать какие ограничения наложены на программу.

## **Оценка**

Итоговая оценка складывается из результатов прогона на тестовых данных, результатов ручного тестирования GUI, полноты реализации задания и, возможно, анализа исходных текстов программы.

# Волга ИТ - 2015. Прикладное Программирование

## Очный тур, второй этап



### Введение

Ага! Все старо как мир! Оказывается, ничего нового изобретать не надо, и все, что нам нужно — вставить в конец письма правдоподобный текст. Нет, вырезанный из классики текст нам не подойдет. Нам нужен генератор.

### Постановка задачи

Необходимо улучшить алгоритм первого задания согласно новым требованиям.

### Улучшенный частотный словарь (300 баллов):

Необходимо найти 100 самых употребляемых словосочетаний. Словосочетание в данном задании — комбинация из одного или нескольких слов без знаков препинания и без учета регистра. Если некоторое словосочетание является частью другого словосочетания и оба одинаково часто встречаются, то словосочетание меньше длины необходимо отбросить.

### Пример ввода:

Show me your flowchart and conceal your tables, and I shall continue to be mystified. Show me your tables, and I won't usually need your flowchart; it'll be obvious.

### Пример вывода:

your	4
and	3
show me your	2
<i>Остальные словосочетания...</i>	

### Генератор текста (300 баллов):

Необходимо сгенерировать новый текст так, что бы для каждой пары подряд идущих слов соответствовала точно такая же пара подряд идущих слов в одном из исходных текстов. Слово в данном задании — комбинация символов со знаками препинания и с учетом регистра. Текст должен начинаться с большой буквы и заканчиваться словом с точкой. Количество слов в сгенерированном тексте должно быть больше  $L$  ( $1 \leq L \leq 1023 * 1024$ ), но не более  $L + 1024$ . Текст по возможности должен содержать как можно больше уникальных слов.

### Пример вывода:

Show me your flowchart and I won't usually need your tables, and I shall continue to be mystified. Show me your flowchart and I won't usually need your flowchart; it'll be obvious.

### Текст сгенерирован? (100 баллов):

Необходимо определить сгенерирован ли текст по вышеупомянутому алгоритму или нет.

### Графический интерфейс (150 баллов):

К графическому интерфейсу имеются новые требования:

- Подсветить самые популярные слова и словосочетания.
- Возможность задать параметр  $L$ , сгенерировать и посмотреть сгенерированный текст.
- Возможность добавлять, удалять и редактировать тексты из базы данных.

## Структура базы данных:

База данных дополнительно содержит следующие таблицы:

- Таблица «generator»:
  - id — уникальный числовой идентификатор текста (primary key)
  - length – рекомендуемая длина сгенерированного текста (L)
  - content — сгенерированный программой текст
- Таблица «checker»:
  - id — уникальный числовой идентификатор текста (primary key)
  - content — текст для проверки на сгенерированность
  - generated — заполняется программой значением 1, если текст сгенерирован или 0 в противном случае

Таблица «generator» изначально содержит поля id и length. Для каждой строки данной таблицы должен быть сгенерирован текст и занесен в поле content. В таблице «checker» изначально заполнены значения id и content. Столбец generated изменяется программой.

# Волга ИТ - 2015. Прикладное Программирование

## Очный тур, третий этап

### Введение

Еще немного улучшений и ни один фильтр в мире не справится с таким потоком нескончаемого уникального спама! Муа-ха-ха-ха!!!

### Постановка задачи

Необходимо улучшить алгоритмы второго задания согласно новым требованиям.



### Улучшенный генератор текста (300 баллов):

Необходимо улучшить генератор текста так, что бы в сгенерированном тексте для каждого  $N$  ( $1 \leq N \leq 10$ ) подряд идущих слов соответствовала точно такая же последовательность из  $N$  подряд идущих слов в одном из исходных текстов.

### Пример ввода:

Show me your flowchart and conceal your tables, and I shall continue to be mystified. Show me your tables, and I won't usually need your flowchart; it'll be obvious.

### Пример вывода (N=3):

Show me your tables, and I shall continue to be mystified. Show me your flowchart and conceal your tables, and I won't usually need your flowchart; it'll be obvious.

### Структура базы данных:

Новая таблица «generator2» имеет все столбцы таблицы «generator», а так же дополнительный столбец combination со значением  $N$ .