

СЕМИНАР 4 ДЕКАБРЯ

Во всех задачах постарайтесь сделать функции универсальными, чтобы они могли работать с любыми типами символов.

1. Напишите функцию для удаления пробельных символов на концах строки:

```
string Strip(const string& str);
```

2. Напишите функции для разбиения строки по разделителю (символу, строке, множеству символов):

```
void Split(const string& str, char delim, vector<string>* res);  
void Split(const string& str, const string& delim, vector<string>* res);
```

```
template <typename Inp, typename Out>  
Out Split(const string& str, Inp firstDelim, Inp lastDelim, Out result);
```

а также для объединения строк через разделитель:

```
string Join(const vector<string>& str, const string& delim);
```

3. Напишите семейство шаблонных функций для преобразования строки в другой тип и обратно:

```
template <typename T>  
T FromString(const string& str);
```

```
template <typename T>  
string ToString(const T& value);
```

4. Напишите свой класс `rev_char_traits`, который позволял бы сравнивать строки по их окончаниям: чтобы сравнить две строки, нужно их перевернуть и сравнить лексикографически. Такой класс мог бы понадобиться для работы со специальными «обратными словарями», в которых слова с одинаковыми окончаниями сгруппированы вместе.

5. (Шутка.)

«По результатам исследования одного английского университета, не имеет значения, в каком порядке расположены буквы в слове. Говорят, чтобы прева и послеждая буквы были на месте. Остальные буквы могут следовать в любом беспорядке, всё равно текст читается без проблем. Причиной этого является то, что мы не читаем каждую букву по отдельности, а всё слово целиком.»

Напишите программу, которая случайно переставляла бы буквы в словах, поступающих на стандартный поток ввода, причем в начале и в конце слова от 1 до 3 букв в зависимости от длины слова должны оставаться на месте. Сравните результат случайной перестановки с сортировкой букв внутри слова по алфавиту.