

СЕМИНАР ПО ПРОГРАММИРОВАНИЮ НА C++
22 СЕНТЯБРЯ 2015 Г.

1. Напишите следующие функции для работы с C-строками, аналогичные функциям стандартной библиотеки C `string.h`:

```
// вычисляет длину C-строки str
int strlen (const char * str);

// приписывает вторую C-строку к первой
// возвращает указатель на начало destination
char * strcat (char * destination, const char * source);

// переворачивает C-строку source и записывает результат в destination
// возвращает указатель на начало destination
char * strrev (char * destination, const char * source);

// сравнивает две C-строки
// результат определяется знаком возвращаемого значения
int strcmp (const char * str1, const char * str2);
```

Постарайтесь написать как можно более выразительный код. Считается, что необходимая память для хранения новых строк уже выделена пользователем и `destination` указывает на ее начало.

2. Напишите функцию `int atoi (const char * str)`, переводящую десятичную запись целого числа из C-строки в тип `int`.

3. Напишите функцию `char * itoa (unsigned int value, char * str, unsigned int base)`, записывающую целое число `value` в C-строку `str` в системе счисления по основанию `base`. Функция должна возвращать указатель на начальный элемент заполненной строки. Считается, что память для хранения строки уже выделена пользователем и `str` указывает на ее начало.

4. Напомним, что *n-граммой* строки называется ее подстрока длины *n*. Напишите функцию `void calcNGramms(const string& str, size_t n, map<string, size_t>* const result)`, которая находила бы все *n*-граммы в строке `str` и записывала бы их в ассоциативный массив `*result` с указанием того, сколько раз такая *n*-грамма встретилась.