

Ограничение времени	0.1 секунд
Ограничение памяти	8.0 Мб

Условие задачи

Реализуйте модуль `cstring`, состоящий из следующих функций:

- `size_t Strlen(const char* str);` Вычисляет длину строки `str` (без учета нулевого символа).
- `int Strcmp(const char* first, const char* second);` Лексикографически сравнивает две строки. Если `first < second`, то возвращает произвольное отрицательное число, если `first > second`, то - положительное. В случае равенства возвращает 0.
- `int Strncmp(const char* first, const char* second, size_t count);` То же, что и предыдущая функция, но сравнивает первые `count` символов строк (либо пока не встретит нулевой символ).
- `char* Strcpy(char* dest, const char* src);` Принимает строку `src` и копирует ее содержимое (вместе с нуль-терминатором) в `dest`. Гарантируется, что `dest` указывает на область памяти, достаточную для хранения `src`. Возвращает `dest`. Замечание: формально, `std::strcpy` дополнительно требует, чтобы последовательности `dest` и `src` не перекрывались (то есть длина строки `src` должна быть больше расстояния между указателями), так как порядок копирования символов не определен. В этом задании допускается более слабое предположение: `dest` не указывает внутрь строки `src`.
- `char* Strncpy(char* dest, const char* src, size_t count);` То же, что и предыдущая функция, но копирует ровно `count` символов из `src`. Если `count` больше размера `src`, то оставшиеся символы заполняются нуль-символами. Возвращает `dest`.
- `char* Strcat(char* dest, const char* src);` Приписывает строку `src` в конец строки `dest` (конкатенация строк). Гарантируется, что `dest` указывает на область памяти, достаточную для хранения итоговой строки. Возвращает `dest`.
- `char* Strncat(char* dest, const char* src, size_t count);` То же, что и предыдущая функция, но добавляет только первые `count` символов из `src` + нулевой символ. Если `count` больше чем длина `src`, то работает так же как и `Strcat(dest, src)`. Возвращает `dest`.
- `const char* Strchr(const char* str, char symbol);` Возвращает указатель на первое появление символа `symbol` в строке `str` (с учетом нулевого символа), либо нулевой указатель, если такого символа нет.
- `const char* Strrchr(const char* str, char symbol);` Возвращает указатель на последнее появление символа `symbol` в строке `str` (с учетом нулевого символа), либо нулевой указатель, если такого символа нет.
- `size_t Strspn(const char* dest, const char* src);` Возвращает количество первых символов `dest`, которые встречаются в строке `src` (без учета нулевого символа).
- `size_t Strcspn(const char* dest, const char* src);` Возвращает количество первых символов `dest`, которые **не** встречаются в строке `src` (без учета нулевого символа).
- `const char* Strpbrk(const char* dest, const char* breakset);` Возвращает указатель на первый элемент `dest`, который лежит в `breakset`, либо нулевой указатель, если такого элемента нет.
- `const char* Strstr(const char* str, const char* pattern);` Осуществляет поиск подстроки `pattern` в строке `str`. Возвращает указатель на начало первого вхождения, либо нулевой указатель, если вхождения найдено не было.