

ЛАБОРАТОРНАЯ РАБОТА

ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРОК

6.1. Объявление строк

Объявление строки аналогично объявлению массива:

char *имя строки* [*размер*]

Важной особенностью является то, что строка должна обязательно заканчиваться нулевым символом `'\0'` – (нуль-терминатор). Длина строки равна количеству символов плюс нулевой символ.

6.2. Функции для работы со строками

Функции для работы со строками сожержатся в библиотеке `string.lib` (подключение: `#include string.h`).

Наиболее часто применяются следующие функции:

char *strcpy(st1, st2) – копирует содержимое строки *st2*, включая нулевой символ, в строку *st1*.

char *strcat(st1, st2) – добавляет справа к строке *st1* содержимое строки *st2*.

int strcmp(st1, st2) – сравнивает содержимое строк *st2* и *st1*. Если *st1* < *st2*, то результат равен `-1`, если *st1* = *st2* – результат равен нулю, если *st1* > *st2* – результат равен `1`.

char *strstr(st1, st2) – возвращает указатель на первое появление подстроки *st2* в строке *st1*.

char *strchr(st, sh) – возвращает указатель на первое появление символа *ch* в строке *st*.

char *strtok(st1, st2) – возвращает указатель на лексему, находящуюся в строке *st1*. При первом вызове функция возвращает указатель на первый символ в *st1*, а после первой лексемы устанавливает нулевой символ. При последующих вызовах функции со значением `NULL` в качестве первого аргумента указатель аналогичным образом переходит к следующим лексемам. После того как закончились все лексемы, указатель устанавливается в `NULL`.

int strlen(st) – возвращает длину строки *st*.

char *strrev(st) – изменяет порядок следования символов в строке на противоположный.

char *strdup(st); – дублирует строку *st*.

char *strlwr(st) – конвертирует символы строки *st* к нижнему регистру.

char *strupr(st) – конвертирует символы строки *st* к верхнему регистру.

int atoi(st) – преобразует строку *st* в число целого типа (*int*).

double atof(st) – преобразует строку *st* в число действительного типа.

char *itoa(a, st, base) – преобразует число целого типа *a* в строку *st* (*base* – основание системы счисления).

char *gcvt(a, dec, st); – преобразует число действительного типа *a* в строку *st*. Значение *dec* указывает на число десятичных разрядов (не более 18).

6.3. Пример выполнения работы

Условие 1. Выделить и вывести на экран все слова произвольной строки. Слова отделяются друг от друга одним или несколькими пробелами.

```
const int size = 128;
char str[size];
cout << "Введите строку: ";
cin.getline(str, size);

char *context = nullptr;
char *token = strtok_s(str, " ", &context);
while (token != nullptr)
{
    cout << token << endl;
    token = strtok_s(nullptr, " ", &context);
}
```

Условие 2. Определить, является ли строка палиндромом, т.е. читается ли она слева направо так же, как и справа налево.

```
char str[] = "А роза упала на лапу Азора";
cout << str << endl;

bool isPalindrome = true;
_strlwr_s(str);

int length = strlen(str);
int i = 0, j = length - 1;
while (i <= j)
{
    while (str[i] == ' ' && i <= j)
    {
        ++i;
    }
    while (str[j] == ' ' && i <= j)
    {
        --j;
    }
    if (str[i] != str[j])
    {
        isPalindrome = false;
        break;
    }
    ++i;
    --j;
}

cout << (isPalindrome ? "Палиндром." : "Не палиндром.") << endl;
```

6.4. Индивидуальные задания

1. Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется от другой одним или несколькими пробелами. Найти количество групп с пятью символами.
2. Дана строка, состоящая из групп нулей и единиц. Найти и вывести на экран самую короткую группу.
3. Дана строка, состоящая из групп нулей и единиц. Подсчитать количество символов в самой длинной группе.
4. Дана строка, состоящая из групп нулей и единиц. Найти и вывести на экран группы с четным количеством символов.
5. Дана строка, состоящая из групп нулей и единиц. Подсчитать количество единиц в группах с нечетным количеством символов.
6. Дана строка, состоящая из букв, цифр, запятых, точек, знаков «+» и «-». Выделить подстроку, которая соответствует записи целого числа.
7. Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков «+» и «-». Выделить подстроку, которая соответствует записи вещественного числа с фиксированной точкой.
8. Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков «+» и «-». Выделить подстроку, которая соответствует записи вещественного числа с плавающей точкой.
9. Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести на экран числа этой строки в порядке возрастания их значений.
10. Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести четные числа этой строки.
11. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Вывести на экран слова этого текста в порядке, соответствующем латинскому алфавиту.
12. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Вывести на экран порядковый номер слова, накрывающего k -ю позицию (если на k -ю позицию попадает пробел, то номер предыдущего слова).
13. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Разбить исходную строку на две подстроки, причем первая длиной k символов (если на k -ю позицию попадает слово, то его следует отнести ко второй строке).
14. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Вывести на экран порядковый номер слова максимальной длины и номер позиции в строке, с которой оно начинается.
15. Дана строка символов, состоящая из произвольного текста, слова разделены пробелами. Вывести на экран порядковый номер слова минимальной длины и количество символов в этом слове.