**ЛАБОРАТОРНАЯ РАБОТА № 9**

**С++: СТРОКИ**

## ТЕОРИЯ И ПОЯСНЕНИЯ

### *Символы*

Символьные константы заключаются в одинарные кавычки: 'q', '4', '&'.

Переменной типа char выделяется 1 байт памяти.

### *Символьные константы с обратным слэшем*

Управляющие символы, которые нельзя ввести с клавиатуры, оформляются как специальные константы с обратным слэшем. Они используются как обычные константы. Например,

char tab = '\t', new\_line = '\n';

В таблице приведены примеры некоторых из этих констант.

|  |  |
| --- | --- |
| Код | Значение |
| \n | Новая строка |
| \t | Горизонтальная табуляция |
| \" | Двойная кавычка |
| \' | Одинарная кавычка |
| \0 | Нулевой символ |
| \\ | Обратный слэш |
| \N | Восьмеричная константа (N — восьмеричное значение) |
| \xN | Шестнадцатеричная константа (N — шестнадцатеричное значение) |

### *Строки*

При обработке текстов удобно иметь доступ не только к отдельным символам, но и к последовательностям символов – строкам. Строка в С – это последовательность символов, заканчивающаяся символом с кодом 0: '\0'**.** Этот символ не учитывается при определении длины строки.

Для работы со строкой как единым объектом имеются стандартные функции, все они используют нуль-символ в качестве ограничителя.

Строковые константы заключаются в двойные кавычки:

"слово", "много слов", "1 2 3 4 5"

Нуль-символ автоматически добавляется в конец строковых констант. Например, строка"много слов" содержит 11 символов, а ее длина равна 10. В программе строковая константа трактуется как константа-указатель на char.

Для хранения последовательности символов используется массив символов. К отдельным символам строки можно обращаться как к элементам массива. Чтобы такой массив обрабатывался как одна строка, в нем обязательно должен присутствовать нуль-символ. Например,

char str\_c[15]; // объявление и определение статического массива,

// str\_c – константа-указатель

В этом массиве можно хранить строки с длиной от 0 до 14 символов. Формировать эти строки можно либо посимвольно, либо с помощью стандартных функций. Пример посимвольной инициализации (7 последних элементов массива не используются):

char let [15] = {'l', 'e', 't', 't', 'e', 'r', 's', '\0'};

Со строками можно работать и с помощью переменных-указателей.

char \*str\_d; // объявление и определение переменной-указателя

Инициализировать указатель можно значением другого указателя или присваиванием адреса динамической памяти, выделенной для строки

str\_d = new char [40];

Для инициализации указателя константой следует использовать ключевое слово const

const char \*str\_c = "Initial string"; //объявление, определение и //инициализация переменной-указателя на константу

Указатель str\_c указывает на постоянное значение типа char. Значение указателя изменять можно, но значение, на которое указывает указатель, изменять нельзя.

### *Массивы строк*

Массив строк – это двумерный массив. Двумерный массив в общем случае рассматривается как массив одномерных массивов. Статический двумерный массив объявляется следующим образом:

тип s [кол-во строк] [кол-во столбцов];

Поскольку в памяти элементы массива располагаются друг за другом построчно, количество столбцов должно быть известно, в противном случае невозможно вычислить адрес начала очередной строки.

Для массива строк количество столбцов должно быть равно максимальной длине строки + 1.

Примеры работы со статическим массивом из 3 или более строк длиной не менее 6:

strcpy(str[2], "three"); // присваивание значения 2-й строке массива

cout << str[2] << endl; // вывод 2-й строки массива

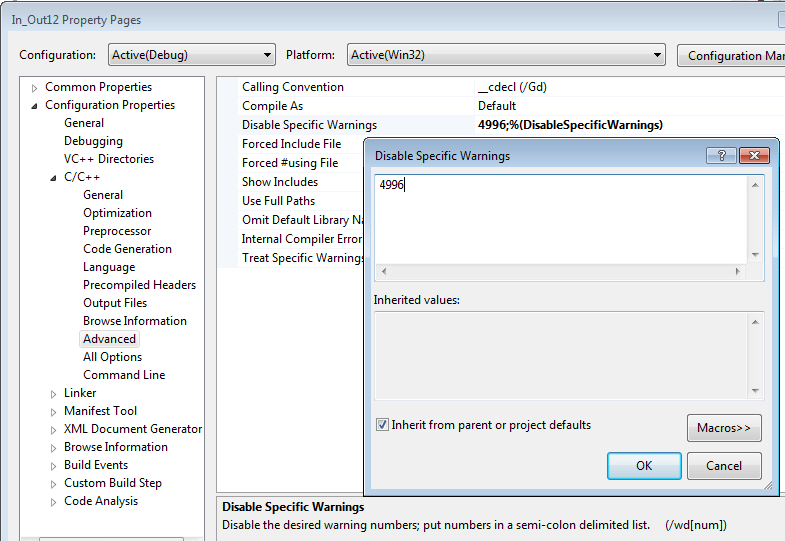
cout << \*str[2] << endl; // вывод начального символа 2-й строки

// массива

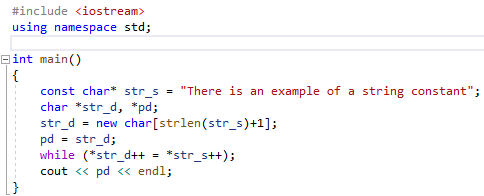
## ЗАДАНИЯ

### УПРАЖНЕНИЕ 1. Написать программу, демонстрирующую использование любых трех функций С++ для работы со строками.

**Замечание**. MS VS настойчиво требует использовать обновленные функции с суффиксом \_s. В MS VS10 это было предупреждением, а в MS VS15 и выше считается ошибкой. Для того чтобы система разрешила использовать прежние функции, нужно выполнить настройку в меню Project – Properties:



### УПРАЖНЕНИЕ 2. Объясните использование типов, переменных и констант в приведенной ниже программе. Определите (без компиляции и выполнения на компьютере), что будет напечатано в результате выполнения этой программы.

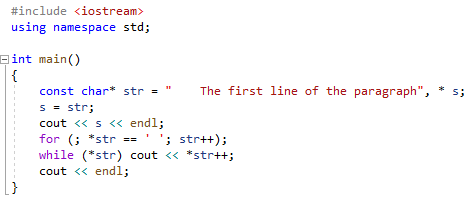


Пояснения.

1. Выполнение оператора присваивания в С++ всегда сопровождается возвращением присвоенного значения, которое записывается в системную память. Если оператор присваивания является частью другого оператора, возвращенное значение может быть использовано для продолжения вычислений. Например, оператор a = b; возвращает значение переменной b, которое не используется. А оператор a = b = c; присваивает переменной b значение переменной c и возвращает значение переменной c, которое далее используется для присваивания переменной a.

2. Целочисленные данные можно использовать в качестве операндов логических выражений. При этом предполагается, что нулевое значение соответствует false, а любое ненулевое значение соответствует true.

### УПРАЖНЕНИЕ 3. Объясните использование типов, переменных и констант в приведенном ниже фрагменте программы. Определите (без компиляции и выполнения на компьютере), что будет напечатано в результате выполнения этого фрагмента программы.



### УПРАЖНЕНИЕ 4. Напишите программу, заполняющую массив строк словами, вводимыми с клавиатуры. Количество слов не более 20, максимальная длина слова – 10. Вывести слова с четными номерами.

***Входные данные***: общее число слов, слова (по одному в строке).

***Результат***: слова из массива печатаются по одному в строках вывода.