

Программирование на языке C++

Символьные строки

Зачем нужны символьные строки?

```
char s[10]; // массив символов
```

- ⊘ элементы массива – отдельные объекты
- сложно работать со строками переменной длины

Хочется:

- строка – единый объект
- длина строки может меняться во время работы программы

```
string s; // символьная строка
```

строка

Символьные строки

Начальное значение:

```
string s = "Привет!";
```

Присваивание:

```
s = "Привет!";
```

Вывод на экран:

```
cout << s;
```



А если массив?

Символьные строки

Ввод с клавиатуры:

```
cin >> s;
```

только до пробела!

```
getline ( cin, s );
```

до перевода строки (Enter)

Отдельный символ:

```
s[4] = 'a';
```



Символы в строку нумеруются с нуля!

Длина строки:

```
int n;
```

```
...
```

```
n = s.size();
```

метод для объектов типа **string**

Символьные строки

Задача: заменить в строке все буквы 'а' на буквы 'б'.

```
#include <iostream>
using namespace std;
main()
{
    string s;
    int i;
    cout << "Введите строку: ";
    getline ( cin, s );
    for ( i = 0; i < s.size(); i++ )
        if ( s[i] == 'a' )
            s[i] = 'б';
    cout << s;
}
```

ЦИКЛ ПО ВСЕМ
СИМВОЛАМ СТРОКИ

Задачи

«А»: Ввести с клавиатуры символьную строку и заменить в ней все буквы «а» на «б» и все буквы «б» на «а» (заглавные на заглавные, строчные на строчные).

Пример:

Введите строку:

ааббААББссСС

Результат:

ббааББААссСС

Задачи

«В»: Ввести с клавиатуры символьную строку и определить, сколько в ней слов. Словом считается последовательности непробельных символов, отделенная с двух сторон пробелами (или стоящая с краю строки). Слова могут быть разделены несколькими пробелами, в начале и в конце строки тоже могут быть пробелы.

Пример:

Введите строку:

Вася пошел гулять

Найдено слов: 3

Задачи

«С»: Ввести с клавиатуры символьную строку и найдите самое длинное слово и его длину. Словом считается последовательности непробельных символов, отделенная с двух сторон пробелами (или стоящая с краю строки). Слова могут быть разделены несколькими пробелами, в начале и в конце строки тоже могут быть пробелы.

Пример:

Введите строку:

Вася пошел гулять

Самое длинное слово: гулять, длина 6

Операции со строками

Объединение (конкатенация):

```
string s, s1, s2;  
s1 = "Привет";  
s2 = "Вася";  
s = s1 + ", " + s2 + "!";
```

"Привет, Вася!"

Срез (подстрока):

```
s = "0123456789";  
s1 = s.substr( 3, 5 ); // "34567"
```

откуда

с какого
символа

СКОЛЬКО
СИМВОЛОВ

5

```
s = "0123456789";  
s1 = s.substr( 3 ); // "3456789"
```

Операции со строками

Удаление:

```
s = "0123456789" ;  
s.erase ( 3, 6 ) ; // "0129"
```

с какого
символа

СКОЛЬКО
СИМВОЛОВ

Вставка:

```
s = "0123456789" ;  
s.insert ( 3, "ABC" ) ; // "012ABC3456789"
```

куда

с какого
символа

ЧТО

Поиск подстроки

```
string s = "Здесь был Вася.";
int n;
n = s.find ( "Вася" ); // 10
```

```
if ( n >= 0 )
    cout << "Слово начинается с s["
        << n << "]\n";
else
    cout << "Слово не найдено.\n";
```



`s.rfind()` – поиск с конца строки!

Пример обработки строк

```
main()
{
    string s, name, name2;
    int n;
    cout << "Введите имя, отчество и фамилию: ";
    getline ( cin, s );
    name = s.substr(0,1) + '.'; // начало имени
    n = s.find(' ');           // найти пробел
    s = s.substr ( n+1 );     // удалить имя
    n = s.find(' ');         // найти пробел
    name2 = s.substr(0,1) + '.'; // начало отчества
    s = s.substr ( n+1 );     // осталась фамилия
    s = s + ' ' + name + name2; // результат
    cout << s;
}
```

Задачи

«А»: Ввести с клавиатуры в одну строку фамилию, имя и отчество, разделив их пробелом. Вывести фамилию и инициалы.

Пример:

Введите фамилию, имя и отчество:

Иванов Петр Семёнович

П.С. Иванов

Задачи

«В»: Ввести адрес файла и «разобрать» его на части, разделенные знаком ' / '. Каждую часть вывести в отдельной строке.

Пример:

Введите адрес файла:

C: /фото/2013/Поход/vasya . jpg

C:

фото

2013

Поход

vasya . jpg

Задачи

«С»: Напишите программу, которая заменяет во всей строке одну последовательность символов на другую.

Пример:

Введите строку:

`(X > 0) and (Y < X) and (Z > Y) and (Z <> 5)`

Что меняем: `and`

Чем заменить: `&`

Результат

`(X > 0) & (Y < X) & (Z > Y) & (Z <> 5)`

Преобразования «строка» – «число»

Из строки в число:

```
string s = "123";  
int N;  
N = atoi ( s.c_str() ); // N = 123
```

«12x3» → 12

в строку
языка Си

```
string s = "123.456";  
float X;  
X = atof ( s.c_str() ); // X = 123.456
```

Преобразования «строка» – «число»

Из числа в строку:

! Идея: направить выходной поток в строку!

```
#include <sstream>
```

строковые потоки

```
ostringstream ss;
```

```
string s;
```

```
int N = 123;
```

```
ss << N;
```

```
s = ss.str(); // s = "123"
```

строковый поток
вывода

из потока в строку

Преобразования «строка» – «число»

Вещественное число в строку:

```
ostringstream ss;  
string s;  
double X = 123.456;  
ss.width(10); // ширина поля  
ss.precision(3); // знаков в дробной части  
ss << X;  
s = ss.str(); // s = " 123.456"
```

Научный формат:

```
ss.str(""); // очистка потока  
ss.width(10); // ширина поля  
ss.precision(6); // знаков в дробной части  
ss << scientific << X; // научный формат  
s = ss.str(); // s = "1.234560E+002"
```

Задачи

«А»: Напишите программу, которая вычисляет сумму трех чисел, введенную в форме символьной строки. Все числа целые.

Пример:

Введите выражение :

12+3+45

Ответ: 60

«В»: Напишите программу, которая вычисляет выражение, состоящее из трех чисел и двух знаков (допускаются только знаки «+» или «-»). Выражение вводится как символьная строка, все числа целые.

Пример:

Введите выражение :

12-3+45

Ответ: 54

Задачи

«С»: Напишите программу, которая вычисляет выражение, состоящее из трех чисел и двух знаков (допускаются знаки «+», «-», «*» и «/»). Выражение вводится как символьная строка, все числа целые. Операция «/» выполняется как целочисленное деление (`div`).

Пример:

Введите выражение :

12*3+45

Ответ: 81

Задачи

«D»: Напишите программу, которая вычисляет выражение, состоящее из трех чисел и двух знаков (допускаются знаки «+», «-», «*» и «/») **и круглых скобок**. Выражение вводится как символьная строка, все числа целые. Операция «/» выполняется как целочисленное деление.

Пример:

Введите выражение :

2 * (3 + 45) + 4

Ответ: 100