



# Работа со строками в C++

**Выполнили:**  
*ст. гр. М40-111Б*

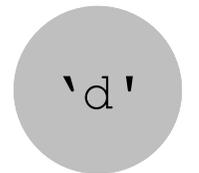
Ходов А. И.

Погула А. Д.

# Символьный тип данных

**Char** – тип данных, предназначенный для хранения одного символа.

```
int main()
{
    char c1 = 'a', c2;
    c1 += 3; //То же самое, что и c1 = c1 + 3;
    //После этого шага значение переменной будет равно символу 'd'
    c2 = 48; //Код символа '0' равен 48
    c1 = '0';
    if (c1 == c2)
        cout << c2 << " " << int(c1); // Вывод: 0 48
    return 0;
}
```



# Строковый тип данных

**Строка** — это последовательность символов. Количество символов в строке называется ее длиной. В C++ для описания строки используется слово *string*. Объявляется строковая переменная следующим образом:

```
string line1;
```

Строке можно присвоить строковую константу, значение другой переменной или выражение:

```
string line1 = "This is a line";           //строковая константа
string line2 = line1;                       //значение другой переменной
string line3 = line1 + " with addition";    //выражение
```

# Строковый тип данных

Пример использования переменной строкового типа:

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string s, ss;    // Объявление переменных s и ss строкового типа
    cin >> s;       // Считывание строки s до первого разделителя
    getline(cin, ss); // Функция getline() считывает все введённые символы
                    // с пробелами до тех пор, пока не будет нажата
// клавиша Enter
    gets(s);        // Полный аналог функции getline()
    cout << s;      // Вывод строки s в консоль
}
```

# Строковый тип данных

В C++ строки имеют больше общего с контейнером *Vector*, нежели с массивами. Отсюда вытекает наличие указателей и возможность использования процедур *sort* и *reverse*.

```
int main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    string ss = "Мокнет Оксана с котенком.";
    reverse(ss.begin(), ss.end()); //Полученная строка:
                                   //".мокнеток с анаско тенкоМ"
    sort(ss.begin(), ss.end());    // Отсортированная строка:
                                   //"MOaaeeekkkkmnnnoostt ."
    return 0;
}
```

# Victor

**Victor** – динамически расширяемый массив, требует подключения библиотеки `<vector>`

```
#include <vector>
...
int main()
{
    vector<char> vec;
    for (int i = 0; i < 5; i++) {
        vec.push_back('a' + i);
    }
    vec.erase(vec.begin() + 2);
    reverse(vec.begin(), vec.end());
    return 0;
}
```



# Библиотека <string>

## Функции библиотеки <string>:

<code>s.length()</code> ;	возвращает длину строки <i>s</i>
<code>s.find(str)</code> ;	возвращает индекс первого вхождения строки <i>str</i> в <i>s</i>
<code>s.rfind(str)</code> ;	возвращает индекс последнего вхождения строки <i>str</i> в <i>s</i>
<code>s.find_first_of(str)</code> ;	возвращает индекс первого вхождения любого символа из <i>str</i> в <i>s</i>
<code>s.find_last_of(str)</code> ;	возвращает индекс последн. вхождения любого символа из <i>str</i> в <i>s</i>
<code>s.substr(k, n)</code> ;	возвращает строку из <i>n</i> символов строки <i>s</i> , начиная с <i>k</i> -ого
<code>s.insert(k, str)</code> ;	вставляет строку <i>str</i> в строку <i>s</i> , начиная с <i>k</i> -ого символа
<code>s.erase(k, n)</code> ;	удаляет <i>n</i> символов в строке <i>s</i> , начиная с <i>k</i> -ого

# Библиотека <string>

Функция `.length()` возвращает длину строки. В данной выводит её в КОНСОЛЬ:

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string ss = "Слава Україні";
    cout << "length: " << ss.length(); //length: 13
    return 0;
}
```

# Библиотека <string>

Функция `.find(str)` (`.rfind(str)`) возвращает индекс первого (последнего) вхождения строки `str` в строку `ss`:

```
#include <string>
using namespace std;
int main() {
    string ss = "Ще не вмерла України і слава, і воля!";
    int a = ss.find("Росія");    //Вхождение не найдено: a = -1
    int u = ss.find("України"); //Вхождение найдено: u = 13
    int e = ss.rfind("і");      //Вхождение найдено: e = 30
    return 0;
}
```

Функция возвращает `-1`, если вхождений не было найдено.

# Библиотека <string>

Функция `.find_first_of(str)` (`.find_last_of(str)`) возвращает индекс первого (последнего) вхождения любого из символов строки `str` в строке `ss`:

```
#include <string>
using namespace std;
int main() {
    string ss = "Рівно 5 хвилин тому, рівно 5 хвилин тому, я був там";
    string p = "0123456789";
    int e = ss.find_first_of(p); // Вхождение найдено: e = 6
    int s = ss.find_last_of(p); // Вхождение найдено: s = 27
    return 0;
}
```

Функция возвращает `-1`, если вхождений не было найдено.

# Библиотека <string>

Функция `.find_first_not_of(str)` возвращает индекс первого вхождения строки `ss` любого символа не содержащихся в строке `str`:

```
#include <string>
using namespace std;
int main() {
    string a = "89992281488 spice smesi soli";
    string b = " 0123456789";
    int c = a.find_first_not_of(b); //Вхождение найдено: c = 12
    return 0;
}
```

Функция возвращает `-1`, если вхождений не было найдено.

# Библиотека <string>

Функция `.substr(k)` возвращает строку, созданную из символов строки `ss`, начиная от `k` до последнего символа.

Функция `.substr(k, n)` возвращает строку, созданную из `n` символов строки `ss`, начиная от `k`.

```
#include <string>
using namespace std;
int main() {
    string ss = "State of Iraq and Syria";
    string s1 = ss.substr(9, 4); // s1 = "Iraq"
    string s2 = ss.substr(18); //s2 = "Syria"
    return 0;
}
```

# Библиотека <string>

Функция `.insert(k, str)` вставляет в строку `ss` начиная с  $k$ -ого символа строку `str`.

```
#include <string>
using namespace std;
int main() {
    string ss = "Ой, нужен , ой";
    ss.insert(10, "ХанаХ");
    //ss = "Ой, нужен ХанаХ, ой"
    return 0;
}
```

# Библиотека <string>

Функция `.erase(k)` удаляет из строки `ss` все символы, начиная с  $k$ -ого.

Функция `.erase(k, n)` удаляет из строки `ss`  $n$  символов, начиная с  $k$ -ого.

```
#include <string>
using namespace std;
int main() {
    string ss = "Бути шаманом-королем, став Всіх шаманів королем.";
    ss.erase(22, 5);
    //ss = "Бути шаманом-королем, Всіх шаманів королем."
    return 0;
}
```

