

Манипуляторы ввода - вывода

Организация рекурсии

```
uint64_t factorial(uint64_t value)
{
    if (value == 1 || value == 0)
        return 1;

    return factorial(value) * factorial(value - 1);
}
```

Организация рекурсии

```
uint16_t positivePow(uint64_t value, uint16_t power, const uint64_t bufValue)
{
    if (power == 0)
        return 1;

    if (power == 1)
        return value;

    return positivePow(value, power - 1, bufValue) * bufValue;
}
```

```
uint16_t positivePow(uint64_t value, uint16_t power)
{
    if (value == 0 && power == 0)
        return 0;

    return positivePow(value, power, value);
}
```

Raw-string

Сырые строки используются для вывода любых последовательностей СИМВОЛОВ

Внешний вид: R “delimiter(raw characters)delimiter”

```
std::cout << R"***(\n\n\n\n\n\s\s  
\s\ada\s\\ \a \  
a)***";  
std::cout << std::endl;
```

Манипуляторы ввода/вывода

Манипуляторы – это вспомогательные функции, которые позволяют управлять потоками ввода / вывода (например, `std::cin`, `std::cout`).

Например,

```
std::cout << std::endl;  
std::cout << std::endl << std::endl;
```

std::hex | std::oct | std::dec

Выводят число в заданной системе счисления.

Находятся в заголовке `<ios>`

```
std::cout << std::hex << 10 << std::endl; // a
std::cout << std::oct << 0xFF << std::endl; // 377
std::cout << std::dec << 0771 << std::endl; // 505
```

std::showbase | std::noshowbase

Отображают выводимую систему
счисления

Находятся в заголовке `<ios>`

```
std::cout << std::hex << 10 << std::endl; // a
```

```
std::cout << std::showbase << 10 << std::endl; // 0xa
```

std::boolalpha | std::noboolalpha

Эти флаги управляют выводом выражений типа `bool`

Находятся в заголовке `<ios>`

```
#include <iostream>
```

```
int main()
{
    bool flag = true;

    std::cout << flag << std::endl; // 1
    std::cout << std::boolalpha << flag << std::endl; // true
    std::cout << std::noboolalpha << flag << std::endl; // 1
}
```


std::showpoint | std::noshowpoint

Управляет отображением плавающей точки (показывать / не показывать)

Находятся в заголовке `<ios>`

```
std::cout << std::showpoint << 1.0 << std::endl; // 1.00000  
std::cout << std::noshowpoint << 1.0 << std::endl; // 1
```

std::showpos | std::noshowpos

Отображает знак '+' перед положительными числами

Находятся в заголовке `<ios>`

```
int value = 4;
```

```
std::cout << value; // 4
```

```
std::cout << std::endl << std::showpos;
```

```
std::cout << value << std::endl; // +4
```

std::skipws | std::noskipws

Поток ввода пропускает / не пропускает пробелы при форматировании
Находятся в заголовке `<ios>`

```
char c1, c2, c3;
```

```
std::cin >> c1 >> c2 >> c3; // "a b c"
```

```
// c1 - a
```

```
// c2 - b
```

```
// c3 - c
```

```
std::cin >> std::noskipws >> c1 >> c2 >> c3; // "a b c"
```

```
// c1 - a
```

```
// c2 - пробел
```

```
// c3 - b
```

std::uppercase | std::nouppercase

Данные флаги управляют регистром шестнадцатеричных чисел и экспоненциальных.

Находятся в заголовке `<ios>`

```
std::cout << std::hex << std::showbase;  
std::cout << std::uppercase << 0x1f << std::endl; // 0x1F  
std::cout << std::nouppercase << 0x1f << std::endl; // 0x1f
```

std::setw(int n)

Старается выравнивать выводимые числа по заданной длине. По умолчанию – по правому краю. Сбрасывается после использования.

Находятся в заголовке `<iomanip>`

```
std::cout << std::setw(10) << 1002 << ' ' << 12 << std::endl; // "      1002 12"
```

std::setfill(CharT c)

Позволяет задать символ для заполнения
Находятся в заголовке `<iomanip>`

```
std::cout << std::setw(10);  
std::cout << std::setfill('#');  
std::cout << std::hex << std::showbase << 0xA << std::endl; //#####0xa
```

std::left | std::right | std::internal

Управляют выравниванием при заданной ширине

Находятся в заголовке `<ios>`

```
std::cout << std::setw(10);  
std::cout << std::setfill('#');  
std::cout << std::hex << std::showbase << std::internal << 0xA << std::endl; //0x#####a  
std::cout << std::setw(10) << std::left << 10 << std::endl; // 0xa#####
```

std::fixed | std::scientific | std::hexfloat std::defaultfloat

Управляют выводом чисел с плавающей запятой.

Находятся в заголовке `<ios>`

std::cout

```
<< "The number 0.01 in fixed: " << std::fixed << 0.01 << '\n'  
<< "The number 0.01 in scientific: " << std::scientific << 0.01 << '\n'  
<< "The number 0.01 in hexfloat: " << std::hexfloat << 0.01 << '\n'  
<< "The number 0.01 in default: " << std::defaultfloat << 0.01 << '\n';
```

Вывод:

```
0.010000  
1.000000e-02  
0x1.47ae14p-7  
0.01
```


std::setprecision(int n)

Устанавливает количество выводимых разрядов в числе с плавающей точкой

Находятся в заголовке `<iomanip>`

```
const long double pi = std::acos(-1.L);
```

```
std::cout << "default precision (6): " << pi << '\n'  
          << "std::setprecision(10): " << std::setprecision(10) << pi << '\n';
```

Вывод:

3.14159

3.141592654

std::flush

«Сбрасывает» данные из буфера выходного потока. Результат может быть не виден на конкретной машине, так как стандарт C++ не регламентирует, когда поток снова «очистит» буфер, но при использовании `std::flush` сброс обязан произойти.

Находятся в заголовке `<ostream>`

std::quoted(const CharT *s) [c++14]

Данный манипулятор выводит текст в кавычках.

Находятся в заголовке `<iomanip>`

```
char string[] = "test";
```

```
std::cout << std::quoted(string); // "test"
```

Некоторые специфичные флаги

http://en.cppreference.com/w/cpp/io/manip/get_money

http://en.cppreference.com/w/cpp/io/manip/put_money

http://en.cppreference.com/w/cpp/io/manip/get_time

http://en.cppreference.com/w/cpp/io/manip/put_time

Работают с вводом /
выводом

денежных валют

Работают с вводом /
выводом

единиц времени