

C: ?????????? ????????

stdio.h

Модуль ввода/вывода.

printf() Форматированный вывод на консоль.

```
#include <stdio.h>

int printf(const char *format, ...);
```

Первый аргумент — шаблон вывода, затем аргументы шаблона (если есть). Функция может принимать переменное число аргументов. При успешном выполнении возвращает кол-во выведенных символов, ошибки — отрицательное значение.

Формат-строка может содержать обычные символы и спецификаторы формата, начинающиеся с символа %. Они указывают, как выводить следующие за строкой аргументы. Примеры спецификаторов:

%c	вывод отдельного символа (char)
%s	вывод строки (char*)
%c	один символ
%d или %i	вывод целого числа со знаком (int)
%u	Целое число без знака (unsigned int) в десятичной системе
%f или %F	вывод числа с плавающей точкой (float, double)
%o	вывод в восьмеричной системе
%x или %X	вывод в шестнадцатеричной системе (строчными и заглавными буквами соответственно)
%%	Символ процента %

Модификаторы типов:

h — для short int (%hd).

l — для long int (%ld) или double (%lf).

L — для long double (%Lf).

Управляющие последовательности внутри строки формата тоже обрабатываются:

\n — перевод на новую строку;

\t — табуляция;

\\ — обратный слеш;
\" — двойная кавычка.

```
printf("Привет, %s!", "мир"); // Выведет «Привет, мир!»
```

Дополнительные возможности форматирования

Флаги:

-	выравнивание по левому краю
+	принудительный вывод знака
#	альтернативное представление для некоторых типов

Ширина поля:

%5d — число будет выведено в поле шириной в 5 символов (по умолчанию — выравнивание по правому краю).

Точность: %.2f — число с плавающей точкой будет выведено с 2 знаками после запятой.

Модификатор размера: %ld (длинное целое), %lld (длинное длинное).

```
printf("%5d", 123); // ' 123'  
printf("%-5d", 123); // '123 ' (выравнивание по левому краю)  
printf("%05d", 12); // '00123' (ведущие нули)  
printf("%+8.2f", 3.14159); // ' +3.14' (знак и ширина)
```

scanf() Считывание данных из стандартного потока ввода (обычно с клавиатуры) и сохранение в переменных. Работает по принципу форматированного ввода: вы указываете шаблон (форматную строку), а функция интерпретирует введенные данные в соответствии с ним.

```
int scanf(const char *format, ...);
```

Возвращает количество успешно прочитанных и присвоенных значений. Если происходит ошибка или конец файла достигнут, возвращается EOF. Однако если например вместо ожидаемой цифры будет введен символ, то он останется в буфере.

Передача адресов переменных

Ключевой момент: функция scanf записывает данные в память, и ей нужно передать адреса переменных, куда эти данные сохраняются. Для этого перед именем переменной ставится амперсанд & (операция взятия адреса).

```
int num;  
scanf("%d", &num); // Передаём адрес переменной num
```

Пример программы

```
#include <stdio.h>

int main() {
    int age;
    printf("Введите возраст: ");
    if (scanf("%d", &age) != 1) {
        printf(stderr, "Ошибка ввода");
        return 1;
    }
    printf("Вы ввели: %d", age);
    return 0;
}
```

Интересные вещи возникают при неправильном вводе.

```
#include <stdio.h>

int main() {
    int age;
    char my;
    printf("Введите возраст: ");
    if (scanf("%d", &age) != 1) {
        printf("Введено: %d\n", age);
        scanf("%c", &my);
        printf("Введено: %c\n", my);
        return 1;
    }
    printf("Вы ввели: %d\n", age);
    return 0;
}
```

Результат:

```
Введите возраст: tre
Введено: 32765
Введено: t
```

- Если переменная не инициализирована, в ней будет какой-то мусор (в данном случае 32765)

- После несоответствия формата ожидаемого ввода, переменная не меняется. То есть в аге остается мусор.
- Ввод попадает в буфер, и только потом считывается scanf.
- Если в буфере будет больше элементов, чем ожидается, то в буфере останется остаток. Он будет считан при последующем вызове функции scanf.
- \n считается символом, поэтому если последовательно получать число и символ, а ввести число и нажать Enter то во вторую переменную попадет \n. То есть буфер после считывания не очищается.

Revision #5

Created 13 July 2026 14:39:48 by Admin

Updated 18 July 2026 15:28:00 by Admin