

Форматоване введення та виведення інформації в C.

Форматоване стандартне введення у мові C здійснює функція **printf**:

```
int printf (char* format, arg1, arg2, . . .) ;
```

Вона повертає кількість успішно виведених аргументів, перетворює, форматує й друкує свої аргументи в стандартному вихідному файлі під керівництвом формату – стрінгу **format**. Останній містить два типи об'єктів: звичайні літери, які безпосередньо копіюються у стандартний вивід, і специфікації форматів перетворень, кожна з яких починається із символу % .

Кількість аргументів функції `printf` має відповідати кількості специфікацій формату. Кожна специфікація завершується спеціальною **літерою-специфікатором**.

Специфікація формату виводу має такий вид:

%[маркер][ширина поля виводу][.точність]**літера-специфікатор**

маркер – це один із знаків :

+ (обов'язковий вивід знаку числа) або

– (притискання до лівого краю поля виводу)

Підкреслені елементи специфікації є обов'язковими.

Таблиця літер-специфікаторів функції

printf

Символ	Тип аргументу	Вид на екрані
<code>d, i</code>	<code>int</code>	десятькове ціле
<code>o</code>	<code>int</code>	вісімкове ціле без знаку
<code>x, X</code>	<code>int</code>	16-ве ціле без знаку
<code>u</code>	<code>int</code>	десятькове ціле без знаку
<code>c</code>	<code>int</code>	СИМВОЛ
<code>s</code>	<code>char*</code>	друкуються символи до ' <code>\0</code> ' або кількість символів визначена точністю
<code>f</code>	<code>double</code>	<code>[-]m.ddddddd</code>
<code>e, E</code>	<code>double</code>	<code>[-]m.dddddddExx</code>
<code>g, G</code>	<code>double</code>	використовує формат <code>%e</code> або <code>%f</code>
<code>p</code>	<code>void*</code>	вказівник
<code>%</code>		друкується символ <code>%</code>

Форматоване стандартне введення в мові C здійснює функція **scanf**:

```
int scanf (char* format, arg1, arg2, ..) ;
```

Вона зчитує символи зі стандартного вхідного потоку, інтерпретує їх згідно зі специфікаціями стрінгу **format** і розсилає результати у свої аргументи, **кожен з яких має бути вказівником**. Повертає кількість успішно введених аргументів. Стрінг **format** містить два види об'єктів: звичайні літери, які, як очікується, **мають з'явитись** у стандартному вводі, і специфікації перетворень, кожна з яких починається із символу **%** . Специфікація формату вводу має такий вид:

%[маркер][ширина поля введення]**літера-специфікатор**

маркер – це знак * (ігнорувати присвоєння)

Підкреслені елементи специфікації є обов'язковими.

Таблиця літер-специфікаторів функції *scanf*

Символ	Тип аргументу	Вид вхідних даних
d	int*	десятькове ціле
i	int*	будь-яке ціле
o	int*	8-ве ціле без знаку
x	int*	16-ве ціле без знаку
u	int*	десятькове ціле без знаку
c	char*	СИМВОЛ
s	char*	ПОСЛІДОВНІСТЬ СИМВОЛІВ
f, e, g	double*	дійсне
%		знак %, ніяке присвоєння не виконується

Зауваження: введення змінних типу **double** в DEV-CPP здійснюється функцією *scanf* некоректно

Форматування потоків введення-виведення в C++

Введення та виведення, пов'язане із стандартними потоками `stdin` та `stdout`, відбувається безформатно, тобто для введення або виведення об'єкту деякого стандартного типу, використовується мінімально необхідна кількість позицій.

Якщо ж виникає необхідність керування станом потоків, то використовуються так звані маніпулятори потоків. Їх визначення містить файл `<iomanip>`, який необхідно підключити командою `#include`. Якщо деякий маніпулятор з'являється в потоці, то він змінює стан потоку до тих пір, поки в потік не буде відправлений інший маніпулятор.

Форматування потоків введення-виведення в C++

Наприклад, за замовчуванням, значення цілих типів читаються та записуються у десятковому форматі. Маніпулятори **hex**, **oct**, **dec** змінюють цілий формат відповідно на 16-ковий, 8-ковий, 10-ковий. Маніпулятори **showbase**, **nshowbase** встановлюють та відмінюють виведення перед числом початкових символів 0 або 0x для позначення основи системи числення. Маніпулятори **uppercase**, **lowercase** визначають вигляд 16-кових цифр (відповідно використовуються великі чи маленькі літери для 16-кових цифр). При введенні та виведенні дійсних значень за замовчуванням діє маніпулятор **fixed** (формат з фіксованою десятковою крапкою). Маніпулятор **scientific** змінює цей формат на формат з рухомою десятковою крапкою. Маніпулятор **setprecision (n)** або функція-член **precision (n)** визначають точність **n** знаків (за замовчуванням точність 6).