

Система типов

1) `int` ($-32767 \dots 32767$) число с таким диапазоном но по факту 32 bit
6 не в счет `ptrdiff_t` ($-32768 \dots 32767$) - диапазон до `ptrdiff_t`

2) `short (int)` (можно без `int`) - не больше чем `int` - 16 bit

3) `long (int)` (можно без `int`) - на `Windows`: 32 bit
на `Unix`: 32-bit на `x32`
64-bit на `x64`

4) `long long (int)` - 64 bit

5) `char` - 8 bit (типа байт) уже то может быть 32 bit

`byte` - мин. адресная единица

`Signed char` $\rightarrow (-128 \dots 127)$

`unsigned char` $\rightarrow (0 \dots 255)$

! можно слева/справа `unsigned` с тем же размером но в ноль

6) `__bool` на C 8 bit т.к. меньше не использовать (у л. бита нет адреса)

можно `bool` при `#include <stdbool.h>` + добавить `True` и `False`

Это все без указания ф.ч. файлов

7) `size_t` / `ptrdiff_t` при `#include <stddef.h>`

`size_t` - беззнаковый тип (где указание по ссылке) `ptrdiff_t` - знаковый тип (где указание по ссылке) `ptrdiff_t` (32 bit на 32 бит. кге и 64)

`ptrdiff_t` - знаковый адрес

2) при `<stdint.h>`:

• `intX_t`, $X: 8, 16, 32, 64$

знаковый
без \rightarrow `uintX_t`, \nearrow

3) ^{формат C23} `_BitInt(X)`, X -подобное кон-во для которого предоставляется компилятор (signed и unsigned)

как писать/читать и константы:

1) X :

$X = 3$; можно $3u$; ^{unsigned int} или $3l$; ^{long} или $3llu$; и т.д.

↑
значения которые помещаются в int/long/long long

2) `_BitInt`:

$X = 3wb$; или $3uwb$;

3) `printf("%i", X)`
^{или}
`%d` ← точно со знаком

без знака: `%u`

`%x` выведет в 16-ой системе

^и
`%X` выведет с прописными и маленькими буквами

если тип: Short то ко всему `%h{...}`
char `-- -- -- --` `%hh{...}`
long `-- -- -- --` `%l{...}`
long long `-- -- -- --` `%ll{...}`
size_t `-- -- -- --` `%z{...}`
ptrdiff_t `-- -- -- --` `%t{...}`

intX_t то: `#include <inttypes.h>`

+ `printf("%", space PRI{...}X, X)`

"u/x/X"

4) scanf ("% " SCN{...}X, &X); echo IntX-t

Указ:

int: scanf ("% i", &X);

↑
4 и 4
% d - 10 c.c.

↑
% X - 16 c.c.

X = 20 ≡ X = 0x14 = 1aB, но X = 010

6 C₂₃: X = 0b1011;

5) __int128 небыл (небыл)

Стандарт IEEE-754

1) float = single pr. 32 bit и точность 24 bit. 10^{~38}

2) double = double pr. 64 bit; точность 53 bit; 10^{~308}

3) long double = double pr. / quad pr. / extended pr.

но чтобы задать - double; X = 1.5f; или X = 1.5d;

3 float16 half pr. но не было

Бог:

% f - float, % lf - double, % Lf - long double

Поддержка комплексной арифметики:

_Complex float X = 1.5f - 2.3if;

↑
вещная часть

очень точная, но медленная. За счет малой сложности можно ускорить

в C++ нет unsigned, но есть `const` (\Rightarrow нельзя изменить) но в C++ есть

арифметика из C

`int x = -2;` \leftarrow `unary minus (const)`

не существует отриц. констант

3 int 8 bit, то -128 не есть в int \Rightarrow user в long long

уже не имеет знака укоротить тогда можно

`int x = -2u;`

`int x, y;`

`typedef unsigned int uint;`

\uparrow
создается `uint`

`uint x; \equiv unsigned int x;`