



知识点多总结





- 数据可分为有符号类型和无符号类型。
- 有符号类型（signed）可以储存正数、负数和零。
定义一个有符号char类型变量的例子： signed char char_1 = 'A'；
- 无符号类型（unsigned）可以储存正数和零。
- 无符号类型仅适用于整数类型（包括字符类型）。
定义一个无符号int类型变量的例子： unsigned int int_1 = 1;
- 整数类型有：整型（int）、短整型（short）、长整型（long）。
- 短整型通常在内存中占2个字节，能表示的数字范围是-32768到32767（即 $-2^{15} \sim 2^{15}-1$ ）。
- 定义一个短整型变量例子： short short_1 = 10; // short是short int的缩写
打印一个短整型变量例子： printf("%hd", short_1);



- 在64位系统上，长整型通常在内存中占8个字节，能表示的数字范围是 $-2^{63} \sim 2^{63}-1$ 。
- 定义一个长整型变量例子： `long long_1 = 10; // long是long int的缩写`
打印一个长整型变量例子： `printf("%ld", long_1);`
- 无符号的整数类型有无符号整型（`unsigned int`） 、无符号短整型（`unsigned short`） 、无符号长整型（`unsigned long`）三种。
- 无符号整型能表示的数字范围是 $0 \sim 2^{32}-1$ ， 打印时对应的占位符是`%u`。
- 无符号短整型能表示的数字范围是 $0 \sim 2^{16}-1$ ， 打印时对应的占位符是`%hu`。
- 无符号长整型能表示的数字范围是 $0 \sim 2^{64}-1$ ， 打印时对应的占位符是`%lu`。



- 计算整数类型的表示范围：

假设占n个比特（1字节 = 8比特）：

有符号的情况下，可表示的范围为： -2^{n-1} 到 $2^{n-1} - 1$ ；

无符号的情况下，可表示的范围为：0到 $2^n - 1$ 。

以短整型short为例：

short占2个字节，也就是16比特，所以它可表示的范围就是 $-2^{15} \sim 2^{15} - 1$ 。



整数可分为有符号整数和无符号整数，其中有符号整数就包括正整数和负整数。

题目1、C语言整数不包括 (A)

- A. 带小数点的整数
- B. 正整数
- C. 负整数
- D. 无符号整数



4 byte (32 bit)

题目2、对于无符号整型数据，以下能正确表示其取值范围的是 (A)

- A. 0 到 $2^{32} - 1$
- B. -2^{31} 到 $2^{31} - 1$
- C. 0 到 $2^{16} - 1$
- D. -2^{15} 到 $2^{15} - 1$



(signed) short int

16 bit

题目3、已知 short int 类型变量占用 2 个字节，其取值范围是 $-2^{15} \sim 2^{15}-1$ 。



题目4、定义一个无符号长整型变量 long_unsigned 的语句是
unsigned long long_unsigned ;。