

C\_C 中的整型常识 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/134/2021\\_2022\\_C\\_C\\_\\_\\_E4\\_B8\\_AD\\_E7\\_9A\\_c97\\_134306.htm](https://www.100test.com/kao_ti2020/134/2021_2022_C_C___E4_B8_AD_E7_9A_c97_134306.htm) 很多人对C/C 中的整型不太了解

，导致代码移植的时候出现问题，本人在此总结一下，若有描述错误，请务必指出，谢谢！

- C/C 对整型长度的规定是为了执行效率，将int定义为机器字长可以取得最大的执行速度.
- C/C 中整型包括：int, char 和 enum, C 中还包含bool类型,C99中bool是一个宏，实际为\_Bool.
- C 和 C 对 enum 的规定有所不同，这里不描述.
- 修饰整型正负的有 signed 和 unsigned，对于 int 默认为 signed.
- 修饰 int 大小的有 short 和 long, 部分编译器还扩展了一些更长的整型，比如 long long 和 \_\_int64, C99中增加了long long和unsigned long long.
- int 的长度与 机器字长相同, 16位的编译器上int长16位,32位的编译器上int长32位.
- short int 的长度 小于等于 int 的长度，注意她们可能长度相等，这取决于编译器.
- long int 的长度 大于等于 int 的长度，注意她们可能长度相等，这取决于编译器.
- char 的长度应当可以包容得下一个字符，大部分系统中就是一个字节，而有的系统中可能是4个字节，因为这些系统中一个字符需要四个字节来描述. 来源：www.examda.com
- char 的正负取决于编译器，而编译器的决定取决于操作系统，在不同的编译器中char可能等同于signed char，也可能等同于unsigned char.

总结: a. 出于效率考虑，应该尽量使用int和unsigned int. b. 当需要指定容量的整型时，不应该直接使用short、int、long 等，因为在不同的编译器上她们的容量不相同。此时应该定义她们相应的宏或类型，比如在VC 6.0中，可以如下定义：

typedef unsigned char UBYTE. typedef signed char SBYTE. typedef unsigned short int UWORD. typedef signed short int SWORD. typedef unsigned int UDWORD. typedef signed int SDWORD. typedef unsigned \_\_int64 UQWORD. typedef signed \_\_int64 SQWORD. 然后在代码中使用 UBYTE、SBYTE、UWORD 等，这样当代码移植的时候只需要修改相应的类型即可。定义自己的类型虽然在代码移植的时候只需要修改一处即可，但仍然属于源代码级别的修改，所以 C 2.0 中将这些类型定义在模板中，可以做到代码移植时无需修改代码。 c. 在定义 char 时，一定要加上 signed 或 unsigned，因为她的正负在不同的编译器上并不相同。 d. 不要想当然的以为 char 是 1 字节长，因为她的长度在不同的编译器上并不相同 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)