short <= int <= long

它们具体占用几个字节C语言并没有规定，C语言只做了宽泛的限制：

* short 至少占用2个字节。
* int 建议为一个机器字长。32位环境下机器字长为4字节，64位环境下机器字长为8字节。
* short 的长度不能大于 int，long 的长度不能小于 int。  
  对于32位的 Windows、Linux 和 OS X，short 为2个字节，int 为4个字节，long 也为4个字节。PC和服务器上的32位系统占有率也在慢慢下降，嵌入式系统使用32位越来越多。  
    
  在64位环境下，不同的操作系统会有不同的结果，如下所示（长度以字节计）：

|  |  |  |  |
| --- | --- | --- | --- |
| **操作系统** | **short** | **int** | **long** |
| Win64 | 2 | 4 | 4 |
| 类Unix系统（包括 Unix、Linux、OS X、BSD、Solaris 等） | 2 | 4 | 8 |

如果是unsigned int，那么可以省略 int ，只写 unsigned，例如：

unsigned n = 100;

它等价于：

unsigned int n = 100;

输出无符号数使用%u，代码如下：

1. #include <stdio.h>
2. #include <stdlib.h>
3. int main()
4. {
5. int a=1234;
6. unsigned a1=1234;
7. int b=0x7fffffff;
8. int c=0x80000000; // 0x80000000 = 0x7fffffff + 0x1
9. int d=0xffffffff;
10. unsigned e=0xffffffff;
11. printf("a=%d, a(u)=%u\n", a, a);
12. printf("a1=%d, a1(u)=%u\n", a1, a1);
13. printf("b=%d, b(u)=%u\n", b, b);
14. printf("c=%d, c(u)=%u\n", c, c);
15. printf("d=%d, d(u)=%u\n", d, d);
16. printf("e=%d, e(u)=%u\n", e, e);
17. system("pause");
18. **return** 0;
19. }

输出结果：  
a=1234, a(u)=1234  
a1=1234, a1(u)=1234  
b=2147483647, b(u)=2147483647  
c=-2147483648, c(u)=2147483648  
d=-1, d(u)=4294967295  
e=-1, e(u)=4294967295  
可以发现，无论变量声明为有符号数还是无符号数，只有当以 %u 格式输出时，才会作为无符号数处理；如果声明为 unsigned，却以 d% 输出，那么也是有符号数。

取值范围和数据溢出

short、int、long 占用的字节数不同，所能表示的数值范围也不同。以32位平台为例，下面是它们的取值范围：

|  |  |  |
| --- | --- | --- |
| **数据类型** | **所占字节数** | **取值范围** |
| short | 2 | -32768~32767，即 -215~(215-1) |
| unsigned short | 2 | 0~65535，即 0~(216-1) |
| int | 4 | -2147483648~2147483647，即 -231~(231-1) |
| unsigned int | 4 | 0~4294967295，即0~(232-1) |
| long | 4 | -2147483648~2147483647，即 -231~(231-1) |
| unsigned long | 4 | 0~4294967295，即0~(232-1) |

当数值过大或过小时，有限的几个字节就不能表示，就会发生溢出。发生溢出时，最高位会被截去。请看下面的例子：

1. #include <stdio.h>
2. int main()
3. {
4. unsigned int a = 0x100000000;
5. printf("a=%u\n", a);
6. **return** 0;
7. }

运行结果：  
a=0  
变量 a 为 int 类型，占用4个字节（32位），能表示的最大值为 0xFFFFFFFF，而 0x100000000 = 0xFFFFFFFF + 1，占用33位，已超出 a 所能表示的最大值，会发生溢出，最高位被截去，剩下的32位都是0。也就是说，在 a 被输出前，其值已经变成了 0。

整数的前缀

在程序中是根据前缀来区分十进制、八进制和十六机制的。  
1) 十进制数由 0~9 十个数字组成，没有前缀。例如：

* 合法的十进制数：237、-568、65535、1627；
* 不合法的十进制数：023（不能有前导0）、23D（含有非十进制数码）。  
  2) 八进制数由 0~7 八个数字组成，必须以0开头，即以0作为八进制数的前缀。例如：
* 合法的八进制数：015（十进制为13）、-0101（十进制为-65）、0177777（十进制为65535）；
* 不合法的八进制数：256（无前缀0）、03A2（包含了非八进制数码）。

注意前缀是数字0，而不是字母o。

3) 十六进制数由数字0~9、字母A~F或a~f组成，前缀为0X或0x。例如：

* 合法的十六进制数：0X2A（十进制为42）、-0XA0（十进制为-160)、0xffff（十进制为65535）；
* 不合法的十六进制数：5A（无前缀0X）、0X3H（含有非十六进制数码）。

在C语言中不能直接表示二进制，它没有特定的前缀。

整数的后缀

1) 可以用后缀L或l来表示长整型数。例如：

* 十进制长整型数：158L、358000L；
* 八进制长整型数：012L、077L、0200000L；
* 十六进制长整型数：0X15L (十进制为21)、0XA5L、0X10000L。  
  长整型数158L和基本整型数158 在数值上并无区别，但由于 158L 是长整型数，编译器将为它分配 sizeof(long) 字节的存储空间。  
  2) 可以用后缀U或u来表示无符号数，例如 358u、0x38Au等。  
    
  前缀、后缀可以同时使用以表示各种类型的整数。例如 0XA5Lu 表示十六进制无符号长整型数 A5，其十进制为165。

实际开发中经常使用前缀，但较少使用后缀，因为将整数赋值给变量时就确定了它是否为 long 类型、是否为 unsigned 类型。

各种整数的输出

在使用 printf 输出整数时，不同的控制字符会有不同的输出格式。  
1) 输出 int 使用%d，输出 short 使用 %hd，输出 long 使用 %ld。  
使用 %d 输出 short，或使用 %ld 输出 int、short 时由于不会发生溢出，所以能够正确输出。而使用 %d 输出 long、或使用 %hd 输出 int、long 时可能会发生数据溢出，导致输出错误。请看下面的例子：

1. #include <stdio.h>
2. #include <stdlib.h>
3. int main()
4. {
5. unsigned short a = 100, b = 0x10000;
6. long c = 0x10, d = 0x10000;
7. printf("a=%d, b=%d\n", a, b);
8. printf("c=%hd, d=%hd\n", c, d);
9. system("pause");
10. **return** 0;
11. }

运行结果：  
a=100, b=0  
c=16, d=0  
变量a、b为 unsigned short 类型，占用2个字节，能表示的最大值为 0XFFFF。a 在输出时使用 %d，能容纳的数值比 a 大，自然不会发生溢出。而 b 被赋值 0x10000，0x10000>0xFFFF，在赋值时就已经发生了溢出，其值为 0，所以 %d 也输出 0。  
变量 c、d 为 long 类型，占用4个字节，能表示的最大值为 0XFFFFFFFF，它们在赋值时都没有溢出。当以 %hd 输出时，会截去较高的两个字节，只输出较低两个字节中的内容。c 的值为 0x10，存储在较低的两个字节中，所以 %hd 能够正确输出。而 d 的值为 0x10000，较低的两个字节全部为0，输出时它的值也就为 0。

实际开发中使用 %d 和 %ld 足以，几乎不使用 %hd。  
2) 输出无符号数使用%u。上面已经讲过，不再赘述。  
3) 输出十进制使用%d，输出八进制使用%o，输出十六进制使用%x或%X。如果希望带上前缀，可以加#，例如 %#d、%#o、%#x、%#X。请看下面的例子：

1. #include <stdio.h>
2. #include <stdlib.h>
3. int main()
4. {
5. int a = 100, b = 0270, c = 0X2F;
6. printf("a(d)=%d, d(#d)=%#d\n", a, a);
7. printf("a(o)=%o, d(#o)=%#o\n", b, b);
8. printf("c(x)=%x, c(#x)=%#x, c(X)=%X, c(#X)=%#X\n", c, c, c, c);
9. system("pause");
10. **return** 0;
11. }

运行结果：  
a(d)=100, d(#d)=100  
a(o)=270, d(#o)=0270  
c(x)=2f, c(#x)=0x2f, c(X)=2F, c(#X)=0X2F  
需要说明的是：

* 十进制数没有前缀，所以 %d 和 %#d 的输出结果一样。
* %o、%x、%X 都是以无符号形式输出。