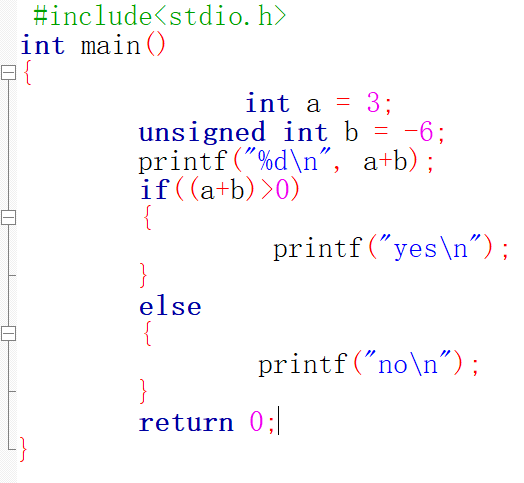
**1、int类型与unsigned int运算的问题**





知识点：

①int类型和unsigned int类型的差别。unsigned int类型比int类型的容量大是因为，unsigned int是无符号类型的，所以最高位不表示正负，而int类型的最高位是表示正负的，1表示负数，0表示正数。

②计算机中负数的存储方式——其补码。在计算机中，负数存储的方式都是以其其补码形式储存，例如：6在计算机的存储形式是0000 0000 0000 0110（32位计算机中），而-6是1000 0000 0000 0110 （32位机下）（最高位是符号位），补码也就是1111 1111 1111 1010（32位计算机中），补码=原码取反+1。

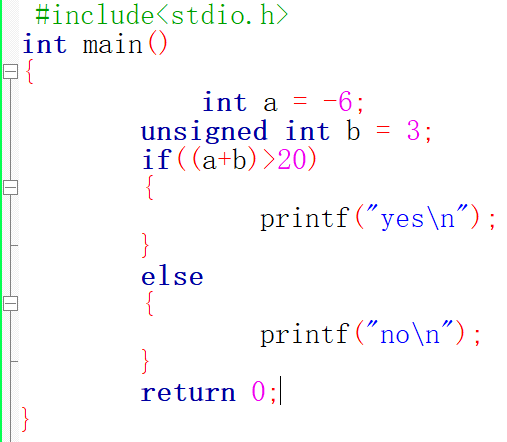
③赋值一个负数给unsigned int类型的变量，它的意思就是将-6这个数的补码形式（1111 1111 1111 1010（32位机下））形式存到一个叫b的空间中，下次我读取这个数的时候，是使用b这个名字，按照b这个名字的类型来读取的，所以当b是unsigned int类型的时候，就不考虑最高位是符号位，则b = 1111 1111 1111 1010（32位机下）（因为是无符号），当b是int类型的时候，考虑最高位是符号位（1为负数，0为正数），当发现是1的时候也就是负数的时候，计算机会把它转换成为原码从而读出是-6。所以一个变量的类型是决定读取这个变量存储的数的方式。

④隐式转换。当unsigned int类型与int类型运算时，如果没有指出转换的类型，它会进行“隐式转换”转换为unsigned int类型，也可以记住，不同类型进行运算时，没指出转换的类型时，它会往容量大的那总类型转换，因为这样可以更加容易避免溢出嘛。

原文链接：<https://blog.csdn.net/gyyu32g/article/details/79089475>

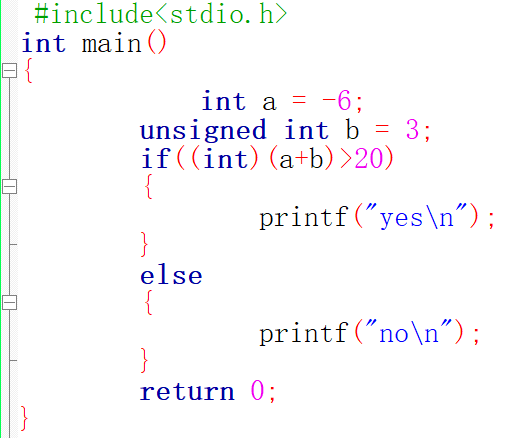
代码分析：

首先，定义了一个 int a = 3 。a的存储以3的二进制形式存储。 unsigned int b = -6 。b的存储是以其补码存储的，也就是1111 1111 1111 1010（32位机下）。当执行a+b的时候，printf函数要求指定打印出%d，所以，a以int（%d）类型读取出来，也就是3；b也是以int（%d）类型读取出来，也就是-6，所以3+（-6）=-3，发现打印结果是-3。而在后面的 if((a+b)>0) 判断中，(a+b)中并没有指定读取的类型，所以就进行隐式转换，a就转成unsigned int类型，也就是3；b转成unsigned int类型，因为b本来的存储是1111 1111 1111 1010（32位机下），转成unsigned int类型时不考虑符号位，所以就直接以补码的形式转化，也就是65530（32位机下），所以a+b=65533>0，输出yes。





改正：





引申：

