各位同学大家好，许多同学在问C++里面隐式类型转换相关的问题，我这里出几个有意思的题给大家思考一下，（这几个题做完你绝对彻底搞懂了）

（1）字面量0x80000000（0x表示16进制）是什么类型？int，unsigned int还是long long?

（2）'9' - '0'得到的结果类型是int还是char？

（3）unsigned short b = -2, c = 3; 请问b+c是什么类型，值为多少？

（4）char c = 28.8 + 2 + 'a' + true; 请问发生了几次类型转换？结果是什么？

【答案】

（1）unsigned int

（2）int

（3）int，65537

（4）4次，ascii=127对应的字符

【解析】

1. 编译器在为整数字面量确定类型的时候遵循一定规则，可以参考这篇[博客](https://www.cnblogs.com/ofnoname/p/15825970.html)。简单来说，

* **对于十进制的字面量**，按照int->long long的顺序找一个表示范围装得下的类型。比如，int的范围是-2147483648（）到2147483647()。那么2147483647就是int，而2147483648就是long long。
* **对于八进制（0开头）或者十六进制（0x开头）的字面量**，按照int -> unsigned int -> long long -> unsigned long long的顺序，寻找一个能把这个数解释成正数的最小类型。比如对于0x80000000，如果确定为int，那么会被解释成，不是正数，那么往上找，如果是unsigned int，那么这个数被解释成，满足要求。

**最后有一点要注意**，编译器为整数字面量确定类型的时候不看负号，它会根据没有负号的那部分确定类型，再计算负号。比如“-2147483648”明明在int的表示范围内，但它的类型是long long而不是int。这是因为编译器会先不看负号，根据“2147483648”确定为long long，然后再取相反数。

1. 字符型，短整型，布尔型这种非标准的整数只要参与运算，自动变成int再算。
2. 注意这里的类型是unsigned short，是无符号数。我们来详细地说一下发生了什么。

* 首先，编译器把-2确定为int，执行转换，变成unsigned short。高精度的整数变成低精度的整数会直接把较低的那几位截取出来。-2作为int，二进制表示为11111111 11111111 11111111 11111110，转成unsigned short截取低16位，就是11111111 11111110。这个由于b是无符号数，所以被解释为65534。
* 后面的3也是同理，先被编译器确定为int类型，再截取低16位，也就是00000000 00000011，（其实也就是3啦）。
* 执行加法之前，由于他们是非标准的整数，所以再全部先转换为int。b被转换为00000000 00000000 11111111 11111110，c被转换为00000000 00000000 00000000 00000011。执行int的加法，这当然不会溢出啦，得到的结果就是65534 +3 = 65537。

1. 从左到右计算，

* 28.8 + 2，把2从int转成double，得到30.8
* 30.8 + ‘a’，把’a’从char转成double，也就是97.0，得到127.8
* 127.8 + true，把true从bool 转成double，也就是1.0，得到128.8
* 把128.8从double转成char，存入c。

为什么把128.8转成char值是127呢？这是浮点数（float/double）转char的特殊规则——把数截断到char能表示的范围，即[-128, 127]里。超过127.0的浮点数转成127，低于-128.0的浮点数转成128。中间的部分就向零取整。

【参考资料】

第三版教材“2.4 算术运算”章节里面的2.4.2，2.4.3章节

前面提到的讲字面量的[博客](https://www.cnblogs.com/ofnoname/p/15825970.html)

接下来附赠一个程序，供大家测试各种表达式的类型、值以及内存中的二进制表达。想测的表达式填在指定位置。

#include <iostream>

#include <bitset>

#include <cxxabi.h>

using namespace std;

int main()

{

  int status;

  // 这里可以定义一些变量，比如我这里定义了一个unsigned short类型的变量b，你可以再定义别的^^

  unsigned short b = -2, d = 3;

  int a = (float)-3.4;

  // 下面这一行的等号后面填写你想要测的表达式，前面的auto还有c不要动

  auto c = b + d + 0x80000000;

  // 下面的代码用来输出变量的类型、值、字节数、内存中的二进制表示，不要动！

  cout << "数据类型：" << abi::\_\_cxa\_demangle(typeid(c).name(), NULL, NULL, &status) << endl;

  cout << "值：" << c << endl;

  cout << "字节数：" << sizeof(c) << endl;

  cout << "内存中的二进制表示：" << bitset<sizeof(c) \* 8>(\*(unsigned long long \*)(&c)) << endl;

  return 0;

}