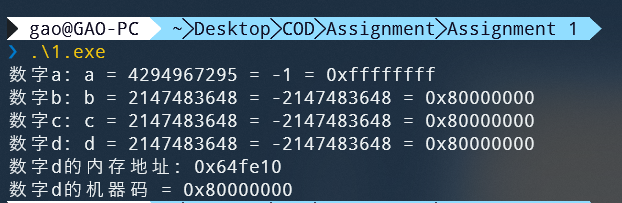
**定点数表示方法报告**

**18081811-高炜哲**

**程序1:**



第一个程序主要目的是将输入的数字, 分别以无符号**十进制整数**, **有符号十进制整数**, **无符号十六进制整数**输出, 同时还输出了d在内存中的地址以及d的机器码.

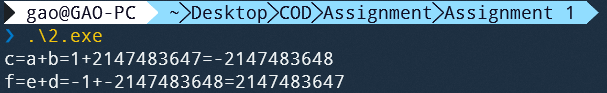
**a:** a为-1,在计算机中以补码储存为1111 1111 1111 1111 1111 1111 1111 1111(0xffffffff), 所以当作无符号整数输出时则是2^32-1 = 4294967295.

**b:** b为2147483648, 由于共32位且,若高1位作为符号位, 所以剩余31位实际上无法表示该数值,因此当以有符号整数输出时,溢出为-2147483648, 即1000 0000 0000 0000 0000 0000 0000 0000(0x80000000), 作为无符号输出时则正常为2147483648.

**c:** c为-2147483648, 在计算机中以补码存储为1000 0000 0000 0000 0000 0000 0000 0000 (0x80000000). 所以作为无符号整数输出时候为2\*31 = 2147483648.

**d:** d被定义为无符号整数-2147483648, 所以作为有符号数输出时正常表示为-2147483648, 作为无符号数输出时为2^31 = 2147483648.

**程序2:**

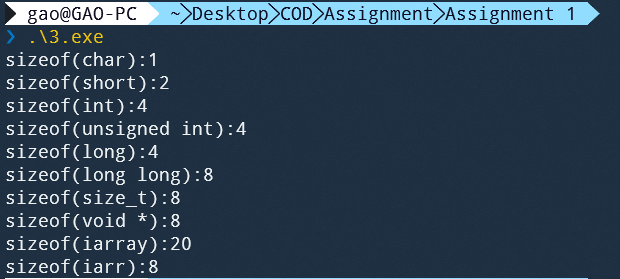


**a+b:** a和b均为有符号int数，最高位32位为符号位，剩余31位最大数字表示为2^31-1 = 2147483647, 所以2147483648上溢，变为-2147483648.

**e+d:** 于上述类似，有符号负数最小被表示为-2147483648(1,000……0), 无法表示

-2147483648-1 = -2147483649，同样上溢，变为2147483647.

**程序3:**



改程序输出了各种类型数据，以及所占的内存空间

**Char字符类型**：在任何机器中都占1字节 = 8位， -128 ~ +127

**Short短整数类型**：占2字节 = 16位，-32767 ~ + 32768

**Int整型**：-2147483648 ~ +2147483647

**Unsigned int无符号整数**：同样占4字节 = 32位，0 ~ 4294967295

**Long长整型**：占4字节 = 32位，与int表示范围相同

**Long long超长整型**：占8字节 = 64位，-9223372036854775808 ~ +9223372036854775807

**Size\_t**：size\_t 类型定义在cstddef头文件中，该文件是C标准库的头文件stddef.h的C++版size\_t是标准C库中定义的，应为unsigned int，其大小足以保证存储内存中对象的大小，所以位8字节 = 64位

**Void\***：指针存放一个内存地址，所以在64位系统中，为8字节 = 64位

**Iarray**：iarray是一个数组，其中存放了5个int类型的数字，所以5\*4=20字节

**Iarr**：iarr同样存放的是一个内存地址，所以同样为8字节，该指针指向数组iarray