

计算机系统原理实验报告

课程名称：____计算机系统原理____实验类型：____上机____

实验项目名称：____整数分析____

学生姓名：____应承峻____专业：____软件工程____学号：____3170103456____

实验日期：____2019____年____2____月____28____日

一、实验描述：

```
#include <iostream>
#include <stdio.h>
int main(int argc, char** argv){
    int x=-2147483648;
    printf("%d, %d", x, -x);
}
```

- (1) 写出运行结果
- (2) 分析为什么会是这样的结果？
- (3) 分析为什么明明一个错误的结果（-x=x）计算机不改正？

二、实验过程：

(1) 实验的运行结果为：-2147483648, -2147483648

(2) 在计算机中，一个 int 类型的变量占 4 个字节（即 32 位），其所能够存储的范围为 $-2^{31} \sim 2^{31}-1$ （即 -2147483648~2147483647）。当对 x 取负时，本应得到 2147483648 的值，但其超过了 int 存放的最大范围，即相当于 $2147483647+1$ ，即：


```
0111 1111 1111 1111 1111 1111 1111 1111
+ 0000 0000 0000 0000 0000 0000 0000 0001
```

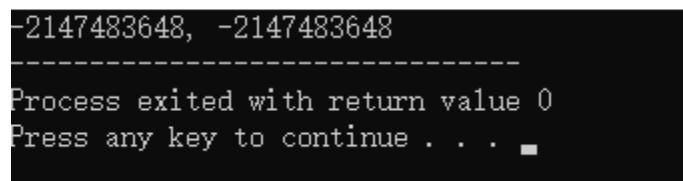
1000 0000 0000 0000 0000 0000 0000 0000

而此时发生了溢出，按补码理解即为-2147483648，因此结果仍然不变

(3) 不改正的主要原因是改正方法较为繁琐

三、上机验证：

 C:\Users\YingChengJun\Desktop\1.exe



```
-2147483648, -2147483648
-----
Process exited with return value 0
Press any key to continue . . .
```

本周的作业“整数分析”是之前学习 C 语言程序设计基础时时常会遇到的一道题，当时由于对计算机中整数的原码、反码和补码表示没有比较深入的理解，所以做 C 程题时每次碰到这种类型的题目时都会非常头疼。这次的作业我首先对 int 类的整数的取值范围进行了计算，发现了取负后 2147483648 会在 int 范围中越界的情况，接着我从网上得知在计算机中整数是以补码的形式存在，然后我再从网上学习了原码、反码和补码的表示和转换的方法，运用这些知识来分析越界后会发生什么样的情况。一开始其实还是有点懵的，算出来的 1000 0000 0000 0000 0000 0000 0000 我怎么也想不通为什么是计算机中最小的整数，后来通过补码计算得到原码和补码相同，然后在网上查到因为 1000 0000 0000 0000 0000 0000 0000 0000 和 0000 0000 0000 0000 0000 0000 0000 都表示 0，所以为了不重复人为规定 1000 0000 0000 0000 0000 0000 0000 0000=-2147483648。所以到此问题得到解决。