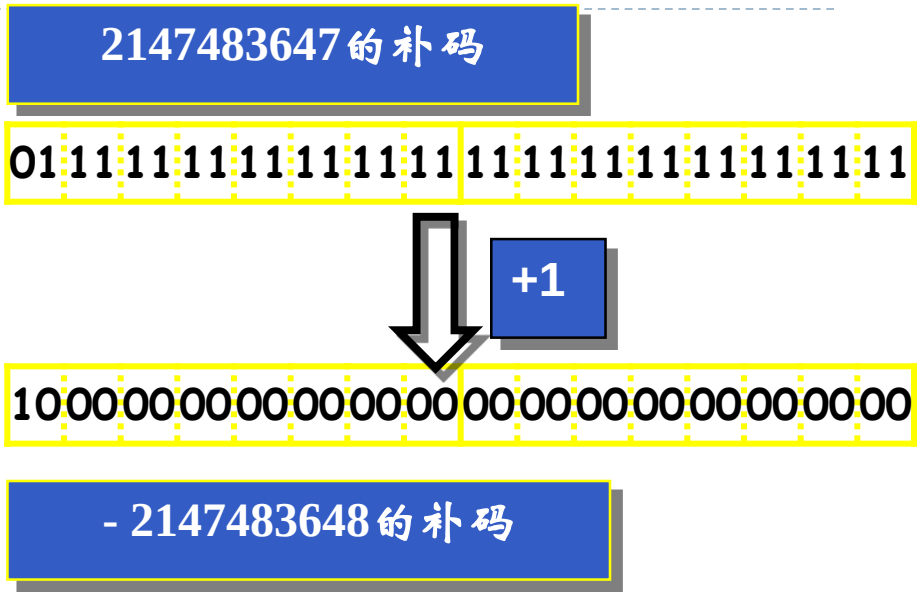
例：整型数据范围溢出示例

```
#include<stdio.h>
int main( )
{
    int a = 2147483647, b = 1;
    printf("%d \n", a + b);
    return 0;
}
```

-2147483648



- 如果将输出格式符%d改成%u（即按unsigned int型数据输出），则结果为2147483648。

5 程序数据描述（I）—补充：整数的补码表示

郭延文

2019级计算机科学与技术系

补充内容

补充阅读材料小结

- 二进制：解决计算机信息存储问题 （知道★★）
- 八、十六进制：简化二进制（了解★）
- 原码：解决正负符号的存储问题（了解★）
- 补码：解决减法运算问题 （知道★★）
- 尾数、指数：解决浮点数存储与计算问题 （了解★）
- BCD码：解决常用十进制数值的存储和运算问题（哦）
- ASCII码：解决常用西文字符的存储问题

（掌握‘0’、‘A’、‘a’的对应十进制！★★）

机器数

补充内容

- 真值：在数值前面用“+”号表示正数，“–”号表示负数的带符号2进制数。
  - +1010111
  - –1010111
- 机器数：用“0”表示符号“+”，用“1”表示符号“–”，即把真值的符号“数值化”，分别用0和1表示（原码）。
  - 8位（1字节）：01010111，11010111
  - 16位（2字节）： 00000000 01010111，10000000 01010111

补充内容

补码

- 补码的简单求法
  - **正数**：同原码
  - **负数**：符号位同原码，其余各位取反，末位加1

真值X：	+1010111	-1010111
$[X]_{原}$(8位)：	01010111	11010111
$[X]_{原}$(16位)：	0000000001010111	1000000001010111
$[X]_{补}$(8位)：	01010111	10101001
$[X]_{补}$(16位)：	0000000001010111	1111111110101001

补充内容

补码的优势

- 数据0的补码只有00000000(8位)
  - 10000000(8位)是-128的补码
- 可将减法变成加法，结果正确

	（0000 0011）补码	3
+	（1111 0011）补码	-13
	（1111 0110）补码	-10

- 8位二进制数：

-128~127

1000 0000~0111 1111

- 32位二进制数：

-2147483648~2147483647

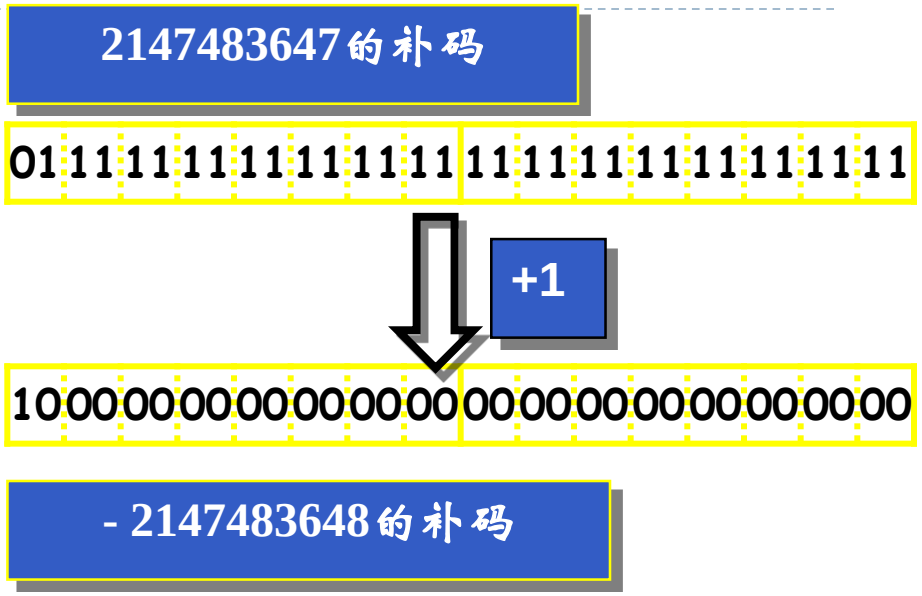
1000 0000 0000 0000 0000 0000 0000 0000~

0111 1111 1111 1111 1111 1111 1111 1111

例：整型数据范围溢出示例

```
#include<stdio.h>
int main( )
{
    int a = 2147483647, b = 1;
    printf("%d \n", a + b);
    return 0;
}
```

-2147483648



- 如果将输出格式符%d改成%u（即按unsigned int型数据输出），则结果为2147483648。

补充内容

- 思考：

  对于内存中的2进制数：

  11111111 11111111 11111111 11111111（4字节）

  显示出来的十进制数是多少？

如果这是一个无符号类型的整型变量，显示为**4294967295**
如果这是一个有符号类型的整型变量（**int**），显示为**-1**

Q & A

为什么机器数表示还不够？ 1. 溢出问题

补充内容

（For simplicity：以8位为例，int其实是32位）

- 不考虑符号

$$
\begin{array}{rl}
 & （1111\ 1111） \\
+ & （0000\ 0001） \\
\hline
 & 1（0000\ 0000）
\end{array}
$$

- 考虑符号

$$
\begin{array}{rl}
 & （0111\ 1111） \\
+ & （0000\ 0001） \\
\hline
 & （1000\ 0000）
\end{array}
$$

为什么机器数表示还不够？
2. 更多的问题

补充内容

- 问题1：0的机器数（原码）不唯一
  - 以8位为例：00000000，10000000

- 问题2：减法运算借位不方便
  - 以8位为例：

	（0000 0011）原码	3
-	（0000 1101）原码	-13
	（?10）原码	

	（0000 0011）原码	3
+	（1000 1101）原码	-13
	（1001 0000）原码	?

- 解决办法：补码