

实验二补充：

	机器数 (十六进制)	真值 (十进制)	OF	SF	CF	AF
x1	0x7fffffff	2147483647	0	1	0	0
y1	0x00000001	1	0	1	0	0
sum_x1_y1	0x80000000	-2147483648	1	1	0	1
diff_x1_y1	0x7ffffffe	2147483646	0	0	0	0
diff_y1_x1	0x80000002	-2147483646	0	1	1	1
x2	0x7fffffff	2147483647	0	1	1	1
y2	0x00000001	1	0	1	1	1
sum_x2_y2	0x80000000	2147483648	1	1	0	1
diff_x2_y2	0x7ffffffe	2147483646	0	0	0	0
diff_y2_x2	0x80000002	2147483650	0	1	1	1

在此题中，x1 表示的是 int 类型的上限，与 y1 相加会发生溢出。

int 类型的表示范围为  $-(1 \ll 31) \sim (1 \ll 31) - 1$ , unsigned 类型的表示范围为  $0 \sim (1 \ll 32) - 1$ ，超过表示范围会发生溢出。

OF: 带符号整型运算指示溢出状态

SF: 有符号整型的最高有效位。(0 指示结果为正，反之则为负)

CF: 无符号整型运算的溢出状态

AF: 算术操作在结果的第 3 位发生进位或借位则将该标志置 1，否则清零

1.

(1)

- 1208248196
- 0
- 如果 %eax 寄存器的二进制序列的最高位为 0, 则 cld 指令就把 %edx 置为 32 个 0 (最后值为 0), 如果 %eax 寄存器的二进制序列最高位为 1, 则 cld 指令将会自填充 %edx 寄存器为 32 个 1 (最后值为 -1)
- 1 idiv 为有符号数的除法指令，将余数存到寄存器 %edx 中

(2)

- 1208248196
- 1
- 0 40000 发生了溢出 (-25536)，能够被整除，余数为 0

2.

```
cmpl $0, 8(%ebp)
jle L2
```

当标志位中的标志意思为小于等于时（由上面的 `cmpl` 影响标志位），跳转到 `.L2`；

```
cmpl $29, 12(%ebp)
```

```
lg .L3
```

当上面两个数的关系判断为大于时，跳转到 `.L3`

要使输入 12 15 的输出结果为 1，把 29 修改成一个小于 15 的数。

学号后四位为 00 76，理由同上。

3.

1) 利用 gdb，分别输入参数：

A: `x = 2 ;`      `result = 14`

B: `x = 6 ;`      `result = 42`

C: `x = 9 ;`      `result = 36`

D: `x = 3 ;`      `result = 30`

2) 学号最后一位为 6，`result = 42`

3)

```
int switch_case(int x){
    int result = 0;
    switch(x){
        case 6:
            result = x*6;
            break;

        case 3:
            result = x*9;
            break;

        case 9:
            result = x* 3;
            break;

        default:
            result = 12;
    }
    return result += x;
}
```