一、

d:8，v：24

大小：32 字节

```
struct {
    double d;
    int i;
    long l;
    void *v;
    short s;
    char c;
} test;
```

24 字节

二、(1) 非永真，如 x=-1，y=-1

(2) 永真，左移等价于乘 2^m

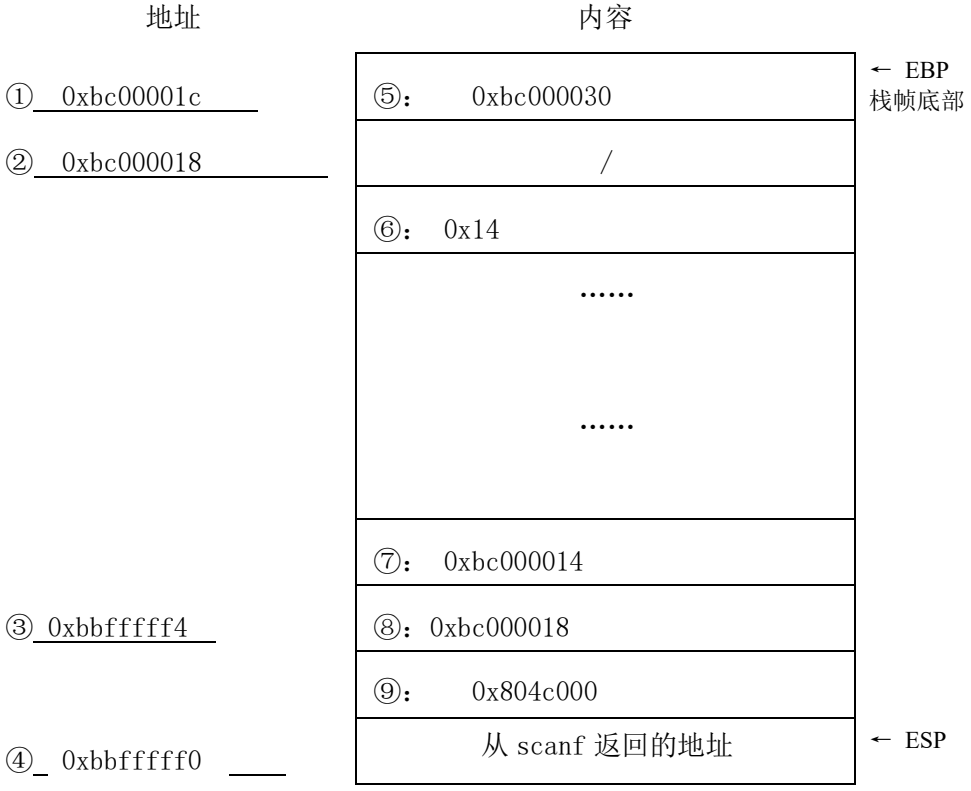
三、int isPositive (int x)

```
{
  return !(!(x^0)|((x>>31)&0x01))
}
```

四、(1)分析执行过程中寄存器 EBP 和 ESP 的内容变化情况（填表，注：带有“/”表项不用填）。

指令及其时刻	EBP 的值	ESP 的值
第 1 行指令执行前	①：0xbc000030	/
第 2 行指令执行后	②：0xbc00001c	③：0xbc00001c
第 3 行指令执行后	/	④：0xbfffff4
第 9 行指令执行完控制即将转向 scanf 时	⑤：0xbc00001c	⑥：0xbfffff0
scanf 执行完控制返回到第 10 行指令时	⑦：0xbc00001c	⑧：0xbfffff4
第 12 行指令执行后	⑨：0xbc000030	⑩：0xbc000020

(2)以下为执行第 9 行指令后 func 的栈帧示意图，写出栈帧中的内容及其地址。



本题：①~④填写单元的地址（每行为一个单元，包含 4 字节，单元的地址指本行 4 字节中最低的地址），⑤~⑨填写单元的内容（单元的内容是指本单元 4 个字节整体解释的值或某种含义，除非特别说明，不用细分成字节讨论）。

⑩ 局部变量 y 所在存储单元的地址是：0xbc000014

五、（1）0x556833b0-0x28=0x55683388

（2）

00	00	00	00	a1	20	C2	04	08	a3	18	C2	04	08
c3	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	8C	33	68	55	05	8d	04	08				

前四个空填 76 a0 8a 4b 也视为对

六、① 0x08049448

② 0x00000104

③ 0x00000448

④ 0x000000e8

⑤ 0x1c

可能的原因是：

.bss 节，文件里没有分配空间，内存里分类了 0x1c 个字节

七、（1）swap，R_386_PC32

重定位前：-4，相对 PC 的偏移

重定位后： swap 首址：0x8048386+0x29=0x80483af，四字节对齐有：0x80483b0

重定位值：0x80483b0-（0x8048386+7-（-4））=0x1f

指向：0x80483b0

（2）sum 在 main 中是弱符号定义，在 app 中是强符号定义。Main 中的 sum 解析到 app 的 sum 上。

Main中对sum赋值-1，对sum的低四字节赋值，但高四字节为0，在app中视为正数，大小为4294967295

另：如果考虑 main 开始的四字节对齐，再讨论后面的取值，也视为正确答案