

汇编语言 作业1

2.9. A. 颜色补: 黑色: 白色

蓝色: 黄色

绿色: 红紫色

蓝绿色: 红色

红色: 蓝绿色

红紫色: 绿色

黄色: 蓝色

白色: 黑色

B. 蓝色 | 绿色 = 蓝绿色

黄色 & 蓝绿色 = 绿色

红色 ^ 红紫色 = 蓝色

2.14. $x \& y: 0x20$

$x \&\& y: 0x01$

$x | y: 0x7F$

$x || y: 0x01$

$\sim x | \sim y: 0xD7$

$!x || !y: 0x00$

$x \& !y: 0x00$

$!x \&\& \sim y: 0x01$

2.15 $!(x \wedge y)$

2.18 A: 736

B: -88

C: 40

D: -48

E: 120

F: 136

G: 504

H: 192

I: -72

2.21	$-2147483647 - 1 == 2147483648U$	无符号数	1
	$-2147483647 - 1 < 2147483647$	有符号数	1
	$-2147483647 - 1U < 2147483647$	无符号数	0
	$-2147483647 - 1 < -2147483647$	有符号数	1
	$-2147483647 - 1U < -2147483647$	无符号数	1

2.23 A.	w	fun1(w)	fun2(w)
	0x00000076	0x00000076	0x00000076
	0x87654321	0x87654321	0x87654321
	0x000000c9	0x000000c9	0xffffffffc9
	0xedcba987	0x00000087	0xffffffff87

B. 函数 fun1 中, 移位运算是在无符号的 word 上进行的, 可看作逻辑移位, 得到的数介于 $0 \sim 255$; 函数 fun2 中, 移位运算是在 word 强制转换为 int 类型后, 变成有符号数后进行的, 可看作算术移位, 得到的数介于 ~~-128~~ $-128 \sim 127$ 。

2.25 原因: 在 for 语句的 $i \leq \text{length} - 1$ 中, length 是 unsigned 类型的, 当 $\text{length} = 0$ 时, $\text{length} - 1$ 将会得到最大的无符号整数, 因此 $i \leq \text{length} - 1$ 始终为真, 循环语句被执行, 最终会访问数组 a 越界后的非法元素, 导致内存错误。

修改: 将 for 语句中的 $i \leq \text{length} - 1$ 改为 $i < \text{length}$

2.26 A. 当字符串 s 比字符串 t 短时, 结果
仍会返回 1

B. 当字符串 s 比字符串 t 短时, 原本期望
 $\text{strlen}(s) - \text{strlen}(t)$ 的结果为负, 但由于
 strlen 的数据类型 size_t 被定义为 unsigned int ,
因此 $\text{strlen}(s) - \text{strlen}(t)$ 反而得到一个非常大
的无符号正整数, 使得结果返回 1

C. 将最后的返回语句改为
`return strlen(s) > strlen(t)`