

## ICS 第二章

### 【整数的表示】

1. 在 x86-64 机器上, 定义 `unsigned int A = 0x123456`。请画出 A 在内存中的存储方式:

...	低地址	A				高地址	...
...		<u>0x</u>	<u>      </u>	<u>0x</u>	<u>      </u>	<u>0x</u>	<u>      </u>

定义 `unsigned short B[2] = {0x1234, 0x5678}`。请画出 B 在内存中的存储方式:

...	低地址	B				高地址	...
...		<u>0x</u>	<u>      </u>	<u>0x</u>	<u>      </u>	<u>0x</u>	<u>      </u>

2. 在 x86-64 机器上, 有下列 C 代码

```
int main() {
    unsigned int A = 0x11112222;
    unsigned int B = 0x33336666;
    void *x = (void *)&A;
    void *y = 2 + (void *)&B;
    unsigned short P = *(unsigned short *)x;
    unsigned short Q = *(unsigned short *)y;
    printf("0x%04x", P + Q);
    return 0;
}
```

运行该代码, 结果为: 0x\_\_\_\_\_。

3. 在 x86-64 机器上, 有下列 C 代码

```
int main() {
    char A[12] = "11224455";
    char B[12] = "11445577";
    void *x = (void *)&A;
    void *y = 2 + (void *)&B;
    unsigned short P = *(unsigned short *)x;
    unsigned short Q = *(unsigned short *)y;
    printf("0x%04x", Q - P);
    return 0;
}
```

运行该代码, 结果为: 0x\_\_\_\_\_。

## 【整数的运算】

4. 在 x86-64 机器上, 有如下的定义:

```
int x = _____;  
int y = _____;  
unsigned int ux = x;  
unsigned int uy = y;
```

判断下列表达式是否等价:

(提示: 减法的运算优先级比按位异或高。布尔运算的结果都是有符号数。)

	表达式 A	表达式 B	等价吗?
(1)	$x > y$	$ux > uy$	Y   N
(2)	$(x > 0) \parallel (x < ux)$	1	Y   N
(3)	$x \wedge y \wedge x \wedge y \wedge x$	$x$	Y   N
(4)	$((x \gg 1) \ll 1) \leq x$	1	Y   N
(5)	$((x / 2) * 2) \leq x$	1	Y   N
(6)	$x \wedge y \wedge (\sim x) - y$	$y \wedge x \wedge (\sim y) - x$	Y   N
(7)	$(x == 1) \&\& (ux - 2 < 2)$	$(x==1) \&\& (!!ux)-2<2)$	Y   N

5. 下列代码的目的是将字符串 A 的内容复制到字符串 B, 覆盖 B 原有的内容, 并输出 "Hello World"; 但实际运行输出是 "Buggy Codes"。尝试找到代码中的错误。

```
int main() {  
    char A[12] = "Hello World";  
    char B[12] = "Buggy Codes";  
    int pos;  
    for (pos = 0; pos - sizeof(B) < 0; pos++)  
        B[pos] = A[pos];  
    printf("%s\n", B);  
}
```

# 【实数的表示】

6. 假设某浮点数格式为 1 符号+3 阶码+4 小数。下表给出了用该格式表达的浮点数 $f = (-1)^S \times M \times 2^E$ 与其二进制表示的关系。完成下表

描述	二进制表示	$M$ (写成分数)	$E$	$f$
负零		-----	-----	-0.0
-----	01000101			
最小的非规格化负数				
最大的规格化正数				
一				1.0
-----				5.5
正无穷		-----	-----	-----

7. 假设浮点数格式 A 为 1 符号+3 阶码+4 小数，浮点数格式 B 为 1 符号+4 阶码+3 小数。回答下列问题。

(1) 格式 A 中有多少个二进制表示对应于正无穷大？

(2) 考虑能精确表示的实数的最大绝对值。A 比 B 大还是比 B 小，还是两者一样？

(3) 考虑能精确表示的实数的最小非零绝对值。A 比 B 大还是比 B 小，还是两者一样？

(4) 考虑能精确表示的实数的个数。A 比 B 多还是比 B 少，还是两者一样？

## 【浮点数的运算】

8. 判断下列说法的正确性

	描述	正确吗?
(1)	对于任意单精度浮点数 a 和 b, 如果 $a > b$ , 那么 $a + 1 > b$ 。	Y    N
(2)	对于任意单精度浮点数 a 和 b, 如果 $a > b$ , 那么 $a + b > b + b$ 。	Y    N
(3)	对于任意单精度浮点数 a 和 b, 如果 $a > b$ , 那么 $a + 1 > b + 1$ 。	Y    N
(4)	对于任意双精度浮点数 d, 如果 $d < 0$ , 那么 $d * d > 0$ 。	Y    N
(5)	对于任意双精度浮点数 d, 如果 $d < 0$ , 那么 $d * 2 < 0$ 。	Y    N
(6)	对于任意双精度浮点数 d, $d == d$ 。	Y    N
(7)	将 float 转换成 int 时, 既有可能造成舍入, 又有可能造成溢出。	Y    N

9. 已知 float 的格式为 1 符号+8 阶码+23 小数, 有下列代码:

```
for (int x = 0; ; x++) {
    float f = x;
    if (x != (int)f) {
        printf("%d", x);
        break;
    }
}
```

其运行结果是 ( )。

A. 死循环

B. 4194305 ( $2^{22} + 1$ )

C. 8388609 ( $2^{23} + 1$ )

D. 16777217 ( $2^{24} + 1$ )

10. 已知 float 的格式为 1 符号+8 阶码+23 小数, 有下列代码:

```
int x = 33554466; //  $2^{25} + 34$ 
int y = x + 8;
for ( ; x < y; x++) {
    float f = x;
    printf("%d ", x - (int)f);
}
```

其运行结果是: \_\_\_\_\_。