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	...
...		0x56	0x34	0x12	0x00	...	

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	...
...		0x34	0x12	0x78	0x56	...	

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_____。【答案：0x5555】

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_____。【答案：0x0303】

【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) || (x < ux)	1	Y N
(3)	x ^ y ^ x ^ y ^ x	x	Y N
(4)	((x >> 1) << 1) <= x	1	Y N
(5)	((x / 2) * 2) <= x	1	Y N
(6)	x ^ y ^ (~x) - y	y ^ x ^ (~y) - x	Y N
(7)	(x == 1) && (ux - 2 < 2)	(x==1) && ((!!ux)-2<2)	Y N

【答】(1)取 x=1，y=-1 即不正确；(2)取 x=-1 即不正确；(3)正确，利用交换律、结合律，以及 x ^ x == 0；(4)正确，即使是对负数；(5)不正确，负奇数该运算向 0 舍入；(6)正确，(~x) - y 也就是 (~x) + (~y) + 1，注意运算优先级；(7)不正确，!!ux 是有符号数。

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);
}
```

【答：sizeof 的结果是无符号数，因此 pos - sizeof(B) < 0 恒假，于是不会进行任何复制。】

【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	10000000	----------	--------	-0.0
----------	01000101	21/16	1	21/8
最小的非规格化负数	10001111	15/16	-2（注意不是-3）	-15/64
最大的规格化正数	01101111	31/16	3	31/2

一	00110000	1	0	1.0
----------	01010110	11/8	2	5.5
正无穷	01110000	----------	--------	-----

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？

【答】 只有一个 01110000

(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？

【答】对于 A 格式，01101111 表示了 31/16*2^3=31/2=15.5，对于 B 格式，01110111 表示了 15/8*2^7=240，因此 B 大。

(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？

【答】 对于 A 格式，00000001 表示了 1/16*2^(-2)=1/64，对于 B 格式，00000001 表示了 1/8*2^(-6)=1/512，因此 A 大。

(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？

【答】A 能精确表达的非负数个数为 7*16=112，B 能精确表达的非负数个数为 15*8=120，因此 B 能精确表达的实数更多。实际上，A 格式表示 NaN 的数比 B 格式多。

【浮点数的运算】

8．判断下列说法的正确性

	描述	正确吗？
(1)	对于任意的单精度浮点数 a 和 b，如果 a > b，那么 a + 1 > b。	Y N
(2)	对于任意的单精度浮点数 a 和 b，如果 a > b，那么 a + b > b + b。	Y N
(3)	对于任意的单精度浮点数 a 和 b，如果 a > b，那么 a + 1 > b + 1。	Y N
(4)	对于任意的双精度浮点数 d，如果 d < 0，那么 d * d > 0。	Y N
(5)	对于任意的双精度浮点数 d，如果 d < 0，那么 d * 2 < 0。	Y N
(6)	对于任意的双精度浮点数 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

【答】(1)正确；(2)取 a = INF、b = FLT_MAX；(3)取 a = 16777220、b = 16777218 即可。这里要特别注意取 a = INF、b = FLT_MAX 不能构成反例，因为 b + 1 因计算精度无法到达 INF；(4)d 取最大的非规格化负数；(5)正确；(6)NAN != NAN；(7)正确。

9．已知 float 的格式为 1 符号+8 阶码+23 小数，则下列程序的输出结果是：

```
for (int x = 0; ; x++) {
```

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

A．死循环

B．4194305 $(2^{22}+1)$

C．8388609 $(2^{23}+1)$

D．16777217 $(2^{24}+1)$

【答】D．表示 16777217 需要 25 位，除了前导 1 以外还要 24 位，float 无法表示。

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：2 -1 0 1 -2 -1 0 1 。

【答】注意 Round to Even 规则。