

ch06 - 编码与位运算

练习 1

1. 下列说法正确的是 (C).
 - A. 在原码中, 符号位为 0 表示负数, 1 表示正数
 - B. 在计算机系统中, 数值一律使用原码来表示和存储
 - C. 正整数的原码、反码、补码表示均是一致的
 - D. 反码是将一个对应的二进制整数原码的 0 和 1 取反
2. 一个整数的 8 位二进制补码是 11111001, 则它的 8 位二进制原码是 (D).
A. 00000111
B. 01111001
C. 11111001
D. 10000111
3. 在 8 位二进制补码中, 11101011 表示的数是十进制下的 (B).
A. 21
B. -21
C. 107
D. -107
4. 数值 12 的 8 位二进制补码为 (B).
A. 1100
B. 00001100
C. 01110100
D. 11110100
5. 在 8 位二进制补码 00111010 的十进制的值为 (A).
A. 58
B. -58
C. 70
D. -70

练习 2

1. 在标准 ASCII 码表中, 数字、小写英文字母和大写英文字母的从小到大的顺序是 (C).
 - A. 数字、小写英文字母、大写英文字母
 - B. 小写英文字母、大写英文字母、数字
 - C. 数字、大写英文字母、小写英文字母
 - D. 大写英文字母、小写英文字母、数字
2. 已知英文字母 h 的 ASCII 码值为 104, 那么字母 H 的 ASCII 码值为 (B).

A. 136

B. 72

C. 71

D. 70

3. 字符 '5' 转换成数字 5 的方法是 (B).

A. '5' - 0

B. '5' - '0'

C. '5' - 5

D. '0' - '5'

4. 阅读程序写结果。

```
#include <iostream>
using namespace std;

int main() {
    char ch = 'D';
    cout << (int)ch << endl;
    return 0;
}
```

68

5. 以下程序的功能为：输入若干个字符（只包含大小写字母，但以 0 结尾，用于标记输入的结束），将该字符串中的所有小写字母变成大写字母并输出。找出其中错误并改正（共 3 处）。

```
#include <iostream>
using namespace std;

int main() {
    char ch;
    while (cin >> ch && ch != 0) {
        if (ch >= 'a' && ch <= 'z') {
            ch += 32;
        }
        cout << ch;
    }
    return 0;
}
```

$ch = ch - 'a' + 'A';$

练习 3

1. 下列运算结果正确的是 (C).

A. $x \wedge 0 = 0$

B. $x \& 0 = x$

C. $x \wedge x = 0$

D. $x | 1 = 1$

2. 52 & 27 的结果为 (C).

A. 0

B. 8

00110100

C. 16

D. 32

3. 52^{27} 的结果为 (B).

A. $(100111)_2$

B. $(101111)_2$

C. $(110001)_2$

D. $(110110)_2$

4. 一般情况下, 一个二进制整数左移两位后, 该数的值为原数的 (A).

A. 4 倍

B. 2 倍

C. $1/2$ 倍

D. $1/4$ 倍

5. 完善程序, 实现功能: 输入一个整数 n , 判断 n 的奇偶性。如果是偶数, 则输出 n 的一半, 如果是奇数, 则输出 n 的两倍。(使用位运算)

```
#include <iostream>
using namespace std;
```

```
int main() {
    int n;
    cin >> n;
    int b = n;
    if (b == 0) { n * 2
        cout << n / 2 << endl;
    } else {
        cout << n * 2 << endl
    }
    cout << n << endl;
    return 0;
}
```

6. 阅读程序回答下列问题。

```
#include <iostream>
using namespace std;
```

```
int main() {
    int n;
    cin >> n;
    cout << (n & (n - 1));
    return 0;
}
```

0000 0101

(1) 输入 5, 输出的结果为 4

(2) 输入 16, 输出的结果为 0

(3) 输入 8, 输出的结果为 0

(4) 以上程序实现的功能为 判断 n 是否为偶数

0000 0000

0000 0000

判断最低位是否为 0.

如果 n 是偶数, 则输出 n / 2
如果 n 是奇数, 则输出 n * 2

课后练习

1. 在 8 位二进制补码 11101010 的十进制的值为 (D)。

- A. 22
- B. -22
- C. 106
- D. -106

2. 完善程序，实现功能：输入一串字符（字符包括大小写字母和数字，以 # 号结尾），实现**大小写字母转换**（大写变小写，小写变大写）、数字原样不变再输出，且不输出 # 号。

```
#include <iostream>
using namespace std;

int main() {
    char ch;
    while (cin >> ch) {
        if (ch >= 'a' && ch <= 'z')
            ch = ch - 'a' + 'A';
        else
            ch = ch - 'A' + 'a';
        cout << ch;
    }
    return 0;
}
```

3. 完善程序，实现功能：输入一个整数 n ，输出 n 的二进制表示（使用位运算）。

```
#include <iostream>

using namespace std;

int res[65];
int main() {
    int n, size = 0;
    cin >> n;
    while (n) {
        res[++size] = n % 2;
        n = n >> 1;
    }
    for (int i = size; i >= 1; i--) {
        cout << res[i];
    }
    return 0;
}
```

4. 以下程序的功能为：输入一个整数 ($0 < n \leq 10^9$)，判断该整数的二进制表示有多少位 1。找出其中的错误并改正（共 2 处）。

```
#include <iostream>
using namespace std;
```

```
int main() {
    int n, num = 1, cnt = 0;
    cin >> n;
    for (int i = 1; i <= 32; i++) {
        cnt += (n & num);
        num = num << 1;
    }
    cout << cnt << endl;
    return 0;
}
```

5. 阅读程序回答下列问题。

```
#include <iostream>
using namespace std;
```

```
int main() {
    int n;
    cin >> n;
    cout << (n & (-n));
    return 0;
}
```

- (1) 输入 5, 输出的结果为 1 4 → 4
- (2) 输入 10, 输出的结果为 2 7 → 1
- (3) 输入 8, 输出的结果为 8
- (4) 以上程序实现的功能为 如果是2的倍数就输出它本身, 否则输出1