

## 6.10 复习题

- 第一题:

1. 请看下面两个计算空格和换行符数目的代码片段:

```
// Version 1
while (cin.get(ch))    // quit on eof
{
    if (ch == ' ')
        spaces++;
    if (ch == '\n')
        newlines++;
}
```

```
// Version 2
while (cin.get(ch))    // quit on eof
{
    if (ch == ' ')
        spaces++;
    else if (ch == '\n')
        newlines++;
}
```

第二种格式比第一种格式好在哪里呢?

版本2执行效率更高，第一种方式中，两个if 的条件判断都会跑一遍，而第二种方式中，如果满足了第一个if条件，else if 的条件将不会执行，效率更高。

- 第二题 在程序清单6.2中，用cht1替换++ch 将发生什么情况呢?

## ◦ 程序清单6.2

**程序清单 6.2 ifelse.cpp**

```
// ifelse.cpp -- using the if else statement
#include <iostream>
int main()
{
    char ch;

    std::cout << "Type, and I shall repeat.\n";
    std::cin.get(ch);
    while (ch != '.')
    {
        if (ch == '\n')
            std::cout << ch;    // done if newline
        else
            std::cout << ++ch;  // done otherwise
        std::cin.get(ch);
    }
    // try ch + 1 instead of ++ch for interesting effect
    std::cout << "\nPlease excuse the slight confusion.\n";
    // std::cin.get();
    // std::cin.get();
    return 0;
}
```

ch 是 char 类型，1 是默认是 int 类型，如果 ch+1 的话，由于类型提升，最后会返回一个 int 类型，以至于 cout 输出的是十进制数字。

而 ++ch 则是 把ch对应的ASCII码加1，类型还是 char 型，最后 cout 输出的是这个ASCII码对应的字符。

- 第三题

### 3. 请认真考虑下面的程序：

```
#include <iostream>
using namespace std;
int main()
{
    char ch;
    int ct1, ct2;

    ct1 = ct2 = 0;
    while ((ch = cin.get()) != '$')
    {
        cout << ch;
        ct1++;
        if (ch = '$')
            ct2++;
        cout << ch;
    }
    cout << "ct1 = " << ct1 << ", ct2 = " << ct2 << "\n";
    return 0;
}
```

假设输入如下（请在每行末尾按回车键）：

**Hi!**

**Send \$10 or \$20 now!**

则输出将是什么（还记得吗，输入被缓冲）？

输出：

```
H$i$!$
$$e$n$d$ $ct1=9, ct2=9
```

注意第二行的开头，开头的\$是因为输入 Hi! 后面敲了一个回车，在while的循环体里，if(ch = '\$') 是一个赋值语句，这个语句一定返回 true，所以ct2++，然后 cout 输出一个 \$，所以第二行的开头是一个 \$。

- 第四题

4. 创建表示下述条件的逻辑表达式:

a. `weight` 大于或等于 115, 但小于 125。

b. `ch` 为 `q` 或 `Q`。

c. `x` 为偶数, 但不是 26。

d. `x` 为偶数, 但不是 26 的倍数。

e. `donation` 为 1000-2000 或 `guest` 为 1。

f. `ch` 是小写字母或大写字母 (假设小写字母是依次编码的, 大写字母也是依次编码的, 但在大小写字母间编码不是连续的)。

```
a. weight >=115 && weight < 125
b. (ch == 'q') || (ch == 'Q')
c. (x%2 == 0) && (x != 26)
d. (x%2 == 0) || (x%26 != 0)
e. (donation >= 1000 && donation <= 2000) || (1 == guest)
f. (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')
```

- 第五题

- 在英语中, “I will not not speak (我不会不说)”的意思与“I will speak(我要说)”相同。在C+中, `!!x` 是否与 `x` 相同呢?

如果 `x` 是 `bool` 类型的话没什么问题, 但如果`x`不是, 就不会相同了, 最后得到的结果不是 `x` 原来的类型而是 `bool`类型。

- 第六题

- 创建一个条件表达式, 其值为变量的绝对值。就是说, 如果变量`x`为正, 则表达式的值为 `x`; 但如果`x`为负, 则表达式的值为 `-x` (这是一个正值)。

```
// 对应的 if-else 语句
if (x >= 0)
    x = x;
else
    x = -x;

// 用条件表达式则是:
(x >= 0) ? x : -x;
```

## 7. 用 switch 改写下面的代码片段:

```
if (ch == 'A')
    a_grade++;
else if (ch == 'B')
    b_grade++;
else if (ch == 'C')
    c_grade++;
else if (ch == 'D')
    d_grade++;
else
    f_grade++;
```

- 第七题

```
switch (ch)
{
    case 'A' : a_grade++; break;
    case 'B' : b_grade++; break;
    case 'C' : c_grade++; break;
    case 'D' : d_grade++; break;
    default : f_grade++;
}
```

- 第八题

- 对于程序清单6-10, 与使用数字相比, 使用字符(如 a 和 c)表示菜单选项和case标签有何优点呢? (提示:想想用户输入 q 和输入 5 的情况。)
- 程序清单6-10:

### 程序清单 6.10 switch.cpp

```
// switch.cpp -- using the switch statement
#include <iostream>
using namespace std;
void showmenu(); // function prototypes
void report();
void comfort();
int main()
{
    showmenu();
    int choice;
```

```
    cin >> choice;
    while (choice != 5)
    {
        switch(choice)
        {
            case 1 :    cout << "\a\n";
                        break;
            case 2 :    report();
                        break;
            case 3 :    cout << "The boss was in all day.\n";
                        break;
            case 4 :    comfort();
                        break;
            default :   cout << "That's not a choice.\n";
        }
        showmenu();
        cin >> choice;
    }
    cout << "Bye!\n";
    return 0;
}

void showmenu()
{
    cout << "Please enter 1, 2, 3, 4, or 5:\n"
          "1) alarm          2) report\n"
          "3) alibi          4) comfort\n"
          "5) quit\n";
}

void report()
{
    cout << "It's been an excellent week for business.\n"
          "Sales are up 120%. Expenses are down 35%.\n";
}

void comfort()
{
    cout << "Your employees think you are the finest CEO\n"
          "in the industry. The board of directors think\n"
          "you are the finest CEO in the industry.\n";
}
```

如果使用数字类型的变量作为 case，那么我们在输入字符的时候，cin 的捕获机制就会出类型不匹配导致 cin 功能禁用的问题，而 while 又在不断循环，整个程序就死循环了。

而如果使用的是 字符 类型变量作为 case，无论输入的是数字还是字母，在cin 捕获的时候就不会出问题。

- 第九题

### 9. 请看下面的代码片段：

```
int line = 0;
char ch;
while (cin.get(ch))
{
    if (ch == 'Q')
        break;
    if (ch != '\n')
        continue;
    line++;
}
```

请重写该代码片段，不要使用 **break** 和 **continue** 语句。

```
int line = 0;
char ch;
while (cin.get(ch) && ch != 'Q')    // cin.get(ch) 确保输入成功，然后再把 break 的条件增加到这里
{
    if(ch == '\n')
        line++;
}
```

## 编程题

---