



switch 嵌套

可以在 switch 语句内使用另一个 switch 语句。

示例程序如下：

```
1. switch(ch1)
2. {
3.     case 'A':
4.         cout << "这个 A 是外部 switch 的一部分";
5.         switch(ch2)
6.         {
7.             case 'A':
8.                 cout << "这个 A 是内部 switch 的一部分";
9.                 break;
10.            case 'B': // 内部 B case 代码
11.        }
12.        break;
13.    case 'B': // 外部 B case 代码
14. }
```



某年某月的天数

【问题描述】

输入 x 和 y ，输出 x 年 y 月有多少天。

说明： x 和 y 均为整数， $1 \leq y \leq 12$

【输入样例】

```
2000 3
```

【输出样例】

```
31
```

【分析】

1、3、5、7、8、10、12 这 7 个月份都是 31 天；

4、6、9、11 这 4 个月份都是 30 天；

如果是闰年，则 2 月份是 29 天，否则 28 天。

【参考程序】

```
#include<iostream>
using namespace std;
int main()
{
    int x, y, day;
    cin >> x >> y;
    switch (y)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            day = 31;
            break;
```



```
    case 4:
    case 6:
    case 9:
    case 11:
        day = 30;
        break;
    case 2:
        if( (x % 400 == 0) || ( (x % 4 == 0) && (x % 100 != 0) ) )
        {
            day = 29;
        }
        else
        {
            day = 28;
        }
        break;
    }
    cout << day << endl;
    return 0;
}
```



四则运算

【问题描述】

一个最简单的计算器支持+, -, *, /四种运算。

输入只有一行：两个参加运算的数和一个操作符 (+, -, *, /)。

输出运算表达式的结果。

考虑下面两种情况：

- (1) 如果出现除数为 0 的情况，则输出：Divided by zero!
- (2) 如果出现无效的操作符（即不为+, -, *, /之一），则输出：Invalid operator!

【输入样例】

```
34+56
```

【输出样例】

```
90
```

【分析】

设 num1、num2 存放两个参加运算的操作数，op 存放操作符。

- ①当 op 为“+”号时，实现加法操作。
- ②当 op 为“-”号时，实现减法操作。
- ③当 op 为“*”号时，实现乘法操作。
- ④当 op 为“/”号时，判断 b 值，如果不为 0，则实现除法操作，如果为 0，则输出：Divided by zero!。
- ⑤当 op 不是上面四种操作符时，输出：“Invalid operator!”。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
```



```
3. int main()
4. {
5.     float num1, num2;
6.     char op;
7.     cin >> num1 >> op >> num2;
8.     switch(op)
9.     {
10.        case '+':
11.            cout << num1 + num2 << endl;
12.            break;
13.        case '-':
14.            cout << num1 - num2 << endl;
15.            break;
16.        case '*':
17.            cout << num1 * num2 << endl;
18.            break;
19.        case '/':
20.            if(num2 != 0)
21.            {
22.                cout << num1 / num2 << endl;
23.            }
24.            else
25.            {
26.                cout << "Divided by zero!" << endl;
27.            }
28.            break;
29.        default:
30.            cout << "Invalid operator!" << endl;
31.    }
32.    return 0;
33. }
```



买笔

【问题描述】

期末来临了，班长小 Q 决定将剩余班费 X （大于等于 8）元钱，用于购买若干支钢笔奖励给一些学习好、表现好的同学。已知商店里有三种钢笔，它们的单价为 6 元、5 元和 4 元。小 Q 想买尽量多的笔（鼓励尽量多的同学），同时他又不想有剩余钱。请帮小 Q 制订出一种买笔的方案。

【分析】

都买 4 元的笔，肯定可以买最多支。因此最多可买的笔为 $\frac{x}{4}$ 支。

由于小 Q 要把钱用完，故我们可以按以下方法将钱用完：

- 若买完 $\frac{x}{4}$ 支 4 元钱的笔，还剩 1 元，则 4 元钱的笔少买 1 支，换成一支 5 元笔；
- 若买完 $\frac{x}{4}$ 支 4 元钱的笔，还剩 2 元，则 4 元钱的笔少买 1 支，换成一支 6 元笔；
- 若买完 $\frac{x}{4}$ 支 4 元钱的笔，还剩 3 元，则 4 元钱的笔少买 2 支，换成一支 5 元笔和一支 6 元笔。

【参考程序】

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     //a、b、c 分别表示在买笔方案中，6 元、5 元和 4 元钱笔的数目
6.     //x、y 分别表示剩余班费和买完最多的 4 元笔后剩的钱
7.     int a, b, c, x, y;
8.     cin >> x;
9.     c = x / 4;    //4 元笔最多买的数目
10.    y = x % 4;    //求买完 c 支 4 元笔后剩余的钱数 y
11.    switch (y)    //判断购买方案
12.    {
13.        case 0:
14.            a = 0;
15.            b = 0;
```



```
16.         break;
17.     case 1:
18.         a = 0;
19.         b = 1;
20.         c -= 1;
21.         break;
22.     case 2:
23.         a = 1;
24.         b = 0;
25.         c -= 1;
26.         break;
27.     case 3:
28.         a = 1;
29.         b = 1;
30.         c -= 2;
31.         break;
32. }
33. cout << a << ' ' << b << ' ' << c << endl;
34. return 0;
35. }
```