东南大学

BJSL0022 - 程序设计基础及语言 I - 12

2026-2027 学年秋季学期

Lab3 控制结构 I

截止日期: 2026 年 11 月 15 日（周日）17：59

学习目标：

1. 掌握“自顶而下、逐步求精”的结构化编程方法
2. 编写 C++ 的 if/else 语句与 while 循环语句
3. 掌握计数器控制的循环与标记控制的循环
4. 理解自增、自减、三元运算符

实验：

Ex1: 使用 C++ 开发个税计算器。现行法律[1]规定，我国的工资、薪金等个人综合所得的个人所得税按全年收入统计，与应纳税所得额（扣除免税收入、减除费用等后）的关系如下：

Table 1: 个人所得税税率表一

级数	全年应纳税所得额	税率	速算扣除数
1	不超过 36000 元的	3%	0
2	超过 36000 元至 144000 元的部分	10%	2520
3	超过 144000 元至 300000 元的部分	20%	16920
4	超过 300000 元至 420000 元的部分	25%	31920
5	超过 420000 元至 660000 元的部分	30%	52920
6	超过 660000 元至 960000 元的部分	35%	85920
7	超过 960000 元的部分	45%	181920

例如，小张每月收入 25000 元，扣除个税起征点 5000 元，应纳税所得收入为 20000 元，则：

1. 一月：税率 3%，税额为 600 元
2. 二月：超过 36000 元的部分为 4000 元，税额为 $16000 * 3\% + 4000 * 10\% = 880$ 元；或使用速算扣除数，全年应纳税额 40000 元属于第二级，因此全年税额为 $40000 * 10\% - 2520 = 1480$ 元，本月应交 $1480 - 600 = 880$ 元
3. 三月：全部收入属于第二级，税额为 $20000 * 10\% = 2000$ 元；或使用速算扣除数，全年应纳税额 60000 元属于第二级，总税额 $60000 * 10\% - 2520 = 3480$ 元，本月应缴纳 $3480 - 1480 = 2000$ 元。
4. 其余月份以此类推

编写程序，计算一年内每次收入所对应的税额。程序循环读入用户的下次实际收入。每次收入的应纳税所得额为 $\max\{0, \text{实际收入} - 5000\}$，即每次 5000 元以内的部分不纳税。程序输出 (1) 本次应纳税额 (2) 全年累计纳税额 (3) 本次税后收入，金额输出固定两位小数。读入-1 则程序结束。

```
(输出样例)
Input next raw income amount (-1 to quit): 25000
Tax: 600.00
Cumulative Tax: 600.00
```

[1] https://fgk.chinatax.gov.cn/zcfgk/c100009/c5193028/content.html

```
After-tax Income: 24400.00

Input next raw income amount (-1 to quit): 25000
Tax: 880.00
Cumulative Tax: 1480.00
After-tax Income: 24120.00

Input next raw income amount (-1 to quit): 25000
Tax: 2000.00
Cumulative Tax: 3480.00
After-tax Income: 23000.00

Input next raw income amount (-1 to quit): -1
```

```cpp
// ex1.cpp
#include <iostream>
/* TODO: include head file for formating float */

using namespace std;

int main() {
    double totalTaxableIncome = 0.0; // tracking the total taxable income
    double totalTax = 0.0; // tracking the total tax
    double rawIncome = 0.0;
    std::cout << "Input next raw income amount (-1 to quit): ";
    std::cin >> rawIncome;

    while (/* TODO: loop condition */) {
        double taxableIncome = rawIncome < 5000 ? 0 : rawIncome - 5000;

        /* TODO: update total taxable income */

        // compute the new total tax using the updated income
        double newTotalTax = 0.0;
        if (totalTaxableIncome <= 36000.0) {
            newTotalTax = totalTaxableIncome * 0.03;
        } else if (totalTaxableIncome <= 144000) {
            newTotalTax = totalTaxableIncome * 0.1 - 2520;
        } else /* TODO: finish the rest */

        /* TODO: compute the tax for this income */

        std::cout << /* TODO: formatting */
                  << "Tax: " << tax << std::endl
                  << "Cumulative Tax: " << newTotalTax << std::endl
                  << "After-tax Income: " << rawIncome - tax << std::endl;

        /* TODO: update loop variables and totalTax */
    }
}
```

问题解决技巧：

1. 使用标记控制的循环不断读入数据，并判断是否终止（不要忘记循环变量更新）
2. 对于每次原始收入，先计算应纳税收入（代码已给出），然后计算全年总共的应纳税所得额（累加），并根据以下公式计算全年应纳税额

$$\text{全年应纳税额} = \text{全年应纳税收入} * \text{对应税率} - \text{速算扣除数}$$

3. 对应税率与速算扣除数根据表中全年总收入对应的行，通过连续的 if ... else if ... 进行判断
4. 分别用 totalTax 记录上一次收入时已纳税总额，用 newTotalTax 记录这一次收入后的应纳税总额，二者相减即是本次应该交的税（不要忘记更新 totalTax）
5. 注意输出格式（不要忘记引用对应头文件）
6. 为了验证算法正确性，应该构造足以覆盖所有 if 分支的输入
7. 可以与在线个税计算器[2]进行对比，确认结果是否正确



思考问题：

1. 修改程序，只输入一次月薪，输出全年 12 个月每个月分别的税后收入

Ex2: 编写程序，输入一个二进制正整数 (由 0 和 1 构成)，输出其十进制表达。

```
(输出样例)
Input a binary number: 10010001
The binary number 10010001 is 145 under base 10
```

```
(输出样例)
Input a binary number: 01010100101
The binary number 1010100101 is 677 under base 10
```

[2]如https://www.gerensuodeshui.cn/

问题解决技巧：

1. 本题不提供模板程序，请根据提示自行解决
2. 可以假设输入足够小（十进制 1000 以内），二进制表达可以整体用`long`读入
3. 从个位开始，通过求模分离出最后一位，此时的 1 代表 1，将对应值（0 或 1）加到十进制数里
4. 将输入的数字整除 10 去掉个位，此时最后一位的 1 代表 2，添加对应值（0 或 2）
5. 继续循环将输入输入整除 10，此时最后一位的 1 代表 4，添加对应值（0 或 4）
6. 直到没有剩余数字（剩下的二进制为 0），则结束循环
7. 注意程序要求输出原始数字，建议循环中不要直接处理输入的数，而是复制一份

思考问题：

1. 如果二进制数很大，超过了`long`所能表示的上限，应该如何将其读入？
2. 能否简单修改代码，实现八进制到十进制的转换？