

TURING

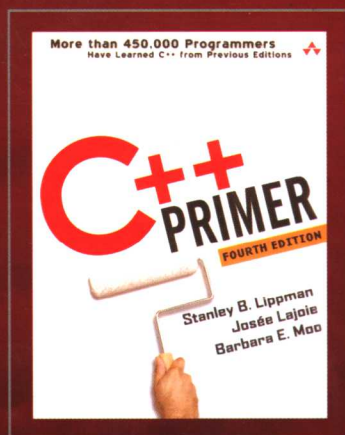
图灵程序设计丛书

C++ Primer (第4版)

习题解答

蒋爱军 李师贤 梅晓勇 编著

- 学习 C++ 的最佳伴侣
- 涵盖所有习题的答案
- 代码均配有详细注释



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

C++ Primer (第4版)

习题解答

蒋爱军 李师贤 梅晓勇 编著

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

C++ Primer (第4版) 习题解答 / 蒋爱军, 李师贤, 梅晓勇著. —北京: 人民邮电出版社, 2007.2
(图灵程序设计丛书)

ISBN 978-7-115-15510-8

I. C... II. ①蒋...②李...③梅... III. C语言—程序设计—解题 IV. TP312-44
中国版本图书馆 CIP 数据核字 (2006) 第 139483 号

内 容 提 要

C++ *Primer* (第4版) 是 C++ 大师 Stanley B. Lippman 丰富的实践经验和 C++ 标准委员会原负责人 Josée Lajoie 对 C++ 标准深入理解的完美结合, 更加入了 C++ 先驱 Barbara E. Moo 在 C++ 教学方面的真知灼见, 是初学者的最佳 C++ 指南, 而且对于中高级程序员, 也是不可或缺的参考书。本书正是这部久负盛名的 C++ 经典教程的配套习题解答。书中提供了 C++ *Primer* (第4版) 中所有习题的参考答案。

本书对使用 C++ *Primer* (第4版) 学习 C++ 程序设计语言的读者是非常理想的参考书。

图灵程序设计丛书

C++ Primer (第4版) 习题解答

-
- ◆ 编 著 蒋爱军 李师贤 梅晓勇
责任编辑 杨海玲
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销
 - ◆ 开本: 800 × 1000 1/16
印张: 21.75
字数: 528 千字 2007 年 2 月第 1 版
印数: 1-5 000 册 2007 年 2 月北京第 1 次印刷

ISBN 978-7-115-15510-8/TP · 5831

定价: 45.00 元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

前 言

C++是一门非常实用的程序设计语言，既支持过程式程序设计，也支持面向对象程序设计，因而也是目前应用极为广泛的一门程序设计语言。

在层出不穷的介绍 C++ 语言的书籍中，*C++ Primer* 是一本广受欢迎的权威之作。强大的作者阵容、全面的内容介绍、新颖的组织方式，使之深受 C++ 爱好者的青睐。本书编者在翻译 *C++ Primer*（第 4 版）的过程中也深深地感受到了这一点。

在学习一门程序设计语言的过程中，亲自动手编写代码是一种极其有效的学习方式，可以对语言的理解和应用达到事半功倍的效果，因此，*C++ Primer*（第 4 版）¹中提供了许多习题，以帮助读者加深对书中内容的理解。

本书试图成为 *C++ Primer*（第 4 版）的配套书籍，根据 *C++ Primer*（第 4 版）中所介绍的内容提供配套习题的解答，书中所给出的“见 xx 节”，均指参见 *C++ Primer*（第 4 版）的相应章节。

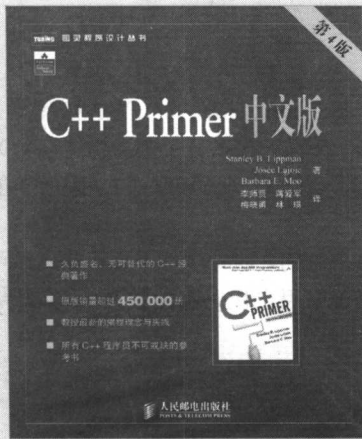
本书中给出的程序均已通过 Microsoft Visual C++ .NET 2003 的编译。源文件（实现文件）以 .cpp 为扩展名，头文件为了与此对应采用 .hpp 为扩展名（而没有采用编译器的默认扩展名 .h）。为了节省篇幅，有些程序中将类的定义与使用类的主函数放在同一实现文件中。包含主函数的源文件根据习题编号命名。大多数模板的定义都没有区分头文件和实现文件（因为编者所用的编译器支持模板的包含编译模型）。另外，使用 Visual C++ .NET 2003 编译器的默认设置会自动连接一些默认库，因此可能有某些所用到的库函数或库类型没有显式指明相应的头文件。使用其他编译器的读者需特别注意，必要时应加上相应的 #include 指示。

衷心希望本书能对使用 *C++ Primer*（第 4 版）学习 C++ 语言的读者有所帮助。

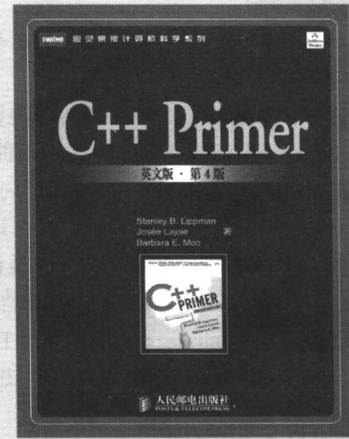
由于编者水平所限，书中不当之处在所难免，恳请读者批评指正。

编 者
2006 年 10 月

1. *C++ Primer*（第 4 版）的中文版及英文影印版均已由人民邮电出版社引进出版。——编者注



书 名: C++ Primer中文版(第4版)
原 书 名: C++ Primer
作 者: Stanley B.Lippman, Josee Lajoie, Barbara E.Moo
译 者: 李师贤 蒋爱军 梅晓勇 林瑛
书 号: 7-115-14554-7
定 价: 99.00元
出版时间: 2006年3月



书 名: C++ Primer(英文版·第4版)
原 书 名: C++ Primer
作 者: Stanley B.Lippman, Josee Lajoie, Barbara E.Moo
书 号: 7-115-15169-7
定 价: 99.00元
出版时间: 2006年10月



书 名: Exceptional C++ Style中文版
原 书 名: Exceptional C++ Style: 40 New Engineering Puzzles, Programming Problems, and Solutions
作 者: Herb Sutter
译 者: 刘未鹏
书 号: 7-115-14225-4
定 价: 39.00元
出版时间: 2005年12月



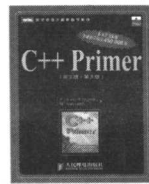
书 名: C++编程规范: 101条规则、准则与最佳实践
原 书 名: C++ Coding Standards: 101 Rules, Guidelines, and Best Practices
作 者: Herb Sutter, Andrei Alexandrescu
译 者: 刘基诚
书 号: 7-115-14205-X
定 价: 35.00元
出版时间: 2005年12月



书 名: C++ 必知必会
原 书 名: C++ Common Knowledge: Essential Intermediate Programming
作 者: Stephen C.Dewhurst
译 者: 荣耀
书 号: 7-115-14101-0
定 价: 29.00元
出版时间: 2005年11月



书 名: C++程序设计(英文版·第5版)
原 书 名: Small C++ How to Program
作 者: H.M.Deitel, P.J.Deitel
书 号: 7-115-14151-7
定 价: 59.00元
出版时间: 2005年12月



书 名: C++Primer(英文版·第3版)
原 书 名: C++ Primer
作 者: Stanley B.Lippman, Josee Lajoie
书 号: 7-115-14056-1
定 价: 69.00元
出版时间: 2005年9月



书 名: C语言程序设计(英文版·第3版)
原 书 名: Programming in C
作 者: Stephen G.Kochan
书 号: 7-115-14763-9
定 价: 55.00元
出版时间: 2006年5月

北京图灵文化发展有限公司 www.turingbook.com
地址: 北京市海淀区西四环北路140号京鼎原商务楼405室
电话: 010-88592340, 88593802 传真: 010-88593803

邮编: 100097
E-mail: contact@turingbook.com

目 录

第 1 章	快速入门	1
第 2 章	变量和基本类型	11
第 3 章	标准库类型	22
第 4 章	数组和指针	36
第 5 章	表达式	53
第 6 章	语句	64
第 7 章	函数	81
第 8 章	标准 IO 库	98
第 9 章	顺序容器	108
第 10 章	关联容器	132
第 11 章	泛型算法	154
第 12 章	类	169
第 13 章	复制控制	189
第 14 章	重载操作符与转换	208
第 15 章	面向对象编程	234
第 16 章	模板与泛型编程	269
第 17 章	用于大型程序的工具	306
第 18 章	特殊工具与技术	323

快速入门

习题1.1

查看所用的编译器文档，了解它所用的文件命名规范。编译并运行本节的main程序。

【解答】

一般而言，C++编译器要求待编译的程序保存在文件中。C++程序中一般涉及两类文件：头文件和源文件。大多数系统中，文件的名称由文件名和文件后缀（又称扩展名）组成。文件后缀通常表明文件的类型，如头文件的后缀可以是.h或.hpp等；源文件的后缀可以是.cc或.cpp等，具体的后缀与使用的编译器有关。通常可以通过编译器所提供的联机帮助文档了解其文件命名规范。

习题1.2

修改程序使其返回-1。返回值-1通常作为程序运行失败的指示器。然而，系统不同，如何（甚至是否）报告main函数运行失败也不同。重新编译并再次运行程序，看看你的系统如何处理main函数的运行失败指示器。

【解答】

笔者所使用的Windows操作系统并不报告main函数的运行失败，因此，程序返回-1或返回0在运行效果上没有什么区别。但是，如果在DOS命令提示符方式下运行程序，然后再键入echo %ERRORLEVEL%命令，则系统会显示返回值-1。

习题1.3

编一个程序，在标准输出上打印“Hello, World”。

【解答】

```
#include<iostream>

int main()
{
    std::cout << "Hello, World" << std::endl;
    return 0;
}
```

习题1.4

我们的程序利用内置的加法操作符“+”来产生两个数的和。编写程序，使用乘法操作符“*”产生两个数的积。

【解答】

```
#include <iostream>

int main()
{
    std::cout << "Enter two numbers:" << std::endl;
    int v1, v2;
    std::cin >> v1 >> v2;
    std::cout << "The product of " << v1 << " and " << v2
              << " is " << v1 * v2 << std::endl;

    return 0;
}
```

习题1.5

我们的程序使用了一条较长的输出语句。重写程序，使用单独的语句打印每一个操作数。

【解答】

```
#include <iostream>

int main()
{
    std::cout << "Enter two numbers:" << std::endl;
    int v1, v2;
    std::cin >> v1 >> v2;
    std::cout << "The sum of ";
    std::cout << v1;
    std::cout << " and ";
    std::cout << v2;
    std::cout << " is ";
    std::cout << v1 + v2 ;
    std::cout << std::endl;

    return 0;
}
```

习题1.6

解释下面的程序段:

```
std::cout << "The sum of " << v1;
          << " and " << v2;
          << " is " << v1 + v2
          << std::endl;
```

这段代码合法吗？如果合法，为什么？如果不合法，又为什么？

【解答】

这段代码不合法。

注意，第1、2、4行的末尾有分号，表示这段代码包含三条语句，即第1、2行各为一个语句，第3、4行构成一个语句。“<<”为二元操作符，在第2、3两条语句中，第一个“<<”缺少左操作数，因此不合法。

在第2、3行的开头加上“std::cout”，即可更正。

习题1.7

编译有不正确嵌套注释的程序。

【解答】

由注释对嵌套导致的编译器错误信息通常令人迷惑。例如，在笔者所用的编译器中编译1.3节中给出的带有不正确嵌套注释的程序：

```
#include <iostream>
/*
 * comment pairs /* */ cannot nest.
 * "cannot nest" is considered source code,
 * as is the rest of the program
 */
int main()
{
    return 0;
}
```

编译器会给出如下错误信息：

```
error C2143: syntax error : missing ';' before '<'
error C2501: 'include' : missing storage-class or type specifiers
warning C4138: '*/' found outside of comment (第6行)
error C2143: syntax error : missing ';' before '{' (第8行)
error C2447: '{' : missing function header (old-style formal list?) (第8行)
```

习题1.8

指出下列输出语句哪些（如果有）是合法的。

```
std::cout << "/*";
std::cout << "*/";
std::cout << /* "*/" */;
```

预测结果，然后编译包含上述三条语句的程序，检查你的答案。纠正所遇到的错误。

【解答】

第一条和第二条语句合法。

第三条语句中<<操作符之后至第二个双引号之前的部分被注释掉了，导致<<操作符的右操作数不是一个完整的字符串，所以不合法。在分号之前加上一个双引号即可更正。

习题1.9

下列循环做什么？sum的最终值是多少？

```
int sum = 0;
for (int i = -100; i <= 100; ++i)
    sum += i;
```

【解答】

该循环求-100~100之间所有整数的和（包括-100和100）。

sum的最终值是0。

习题1.10

用for循环编程，求从50~100的所有自然数的和。然后用while循环重写该程序。

【解答】

用for循环编写的程序如下:

```
#include <iostream>

int main()
{
    int sum = 0;
    for (int i = 50; i <= 100; ++i)
        sum += i;
    std::cout << "Sum of 50 to 100 inclusive is "
              << sum << std::endl;

    return 0;
}
```

用while循环编写的程序如下:

```
#include <iostream>

int main()
{
    int sum = 0, int i = 50;
    while (i <= 100) {
        sum += i;
        ++i;
    }
    std::cout << "Sum of 50 to 100 inclusive is "
              << sum << std::endl;
    return 0;
}
```

习题1.11

用while循环编程, 输出10-0递减的自然数。然后用for循环重写该程序。

【解答】

用while循环编写的程序如下:

```
#include <iostream>

int main()
{
    int i = 10;
    while (i >= 0) {
        std::cout << i << " ";
        --i;
    }

    return 0;
}
```

用for循环编写的程序如下:

```
#include <iostream>
int main()
{
    for (int i = 10; i >= 0; --i)
        std::cout << i << " ";
}
```

```

    return 0;
}

```

习题1.12

对比前面两个习题中所写的循环。两种形式各有何优缺点？

【解答】

在for循环中，循环控制变量的初始化和修改都放在语句头部分，形式较简洁，且特别适用于循环次数已知的情况。在while循环中，循环控制变量的初始化一般放在while语句之前，循环控制变量的修改一般放在循环体中，形式上不如for语句简洁，但它比较适用于循环次数不易预知的情况（用某一条件控制循环）。两种形式各有优点，但它们在功能上是等价的，可以相互转换。

习题1.13

编译器不同，理解其诊断内容的难易程度也不同。编写一些程序，包含本小节“再谈编译”部分讨论的那些常见错误。研究编译器产生的信息，这样你在编译更复杂的程序遇到这些信息时不会陌生。

【解答】

对于程序中出现的错误，编译器通常会给出简略的提示信息，包括错误出现的文件及代码行、错误代码、错误性质的描述。如果要获得关于该错误的详细信息，一般可以根据编译器给出的错误代码在其联机帮助文档中查找。

习题1.14

如果输入值相等，本节展示的程序将产生什么问题？

【解答】

sum的值即为输入值。因为输入的v1和v2值相等（假设为x），所以lower和upper相等，均为x。for循环中的循环变量val初始化为lower，从而val<=upper为真，循环体执行一次，sum的值为val（即输入值x）；然后val加1，val的值就大于upper，循环执行结束。

习题1.15

用两个相等的值作为输入编译并运行本节中的程序。将实际输出与你在习题1.14中所做的预测相比较，解释实际结果和你预计的结果间的不相符之处。

【解答】

运行1.4.3节中给出的程序，输入两个相等的值（例如3,3），则程序输出为：

```
Sum of 3 to 3 inclusive is 3
```

与习题1.14中给出的预测一致。

习题1.16

编写程序，输出用户输入的两个数中的较大者。

【解答】

```
#include <iostream>
```

```
int main()
{
    std::cout << "Enter two numbers:" << std::endl;
    int v1, v2;
    std::cin >> v1 >> v2; // 读入数据

    if (v1 >= v2)
        std::cout << "The bigger number is" << v1 << std::endl;
    else
        std::cout << "The bigger number is" << v2 << std::endl;

    return 0;
}
```

习题1.17

编写程序，要求用户输入一组数。输出信息说明其中有多少个负数。

【解答】

```
#include <iostream>

int main()
{
    int amount = 0, value;

    // 读入数据直到遇见文件结束符，计算所读入的负数的个数
    while (std::cin >> value)
        if (value <= 0)
            ++amount;

    std::cout << "Amount of all negative values read is"
              << amount << std::endl;

    return 0;
}
```

习题1.18

编写程序，提示用户输入两个数并将这两个数范围内的每个数写到标准输出。

【解答】

```
#include <iostream>

int main()
{
    std::cout << "Enter two numbers:" << std::endl;
    int v1, v2;
    std::cin >> v1 >> v2; // 读入两个数

    // 用较小的数作为下界lower、较大的数作为上界upper
    int lower, upper;
    if (v1 <= v2) {
        lower = v1;
        upper = v2;
    } else {
        lower = v2;
        upper = v1;
    }

    // 输出从lower到upper之间的值
    std::cout << "Values of " << lower << "to "
```

```

        << upper << "inclusive are: " << std::endl;
    for (int val = lower; val <= upper; ++val)
        std::cout << val << " ";

    return 0;
}

```

习题1.19

如果上题给定数1000和2000，程序将产生什么结果？修改程序，使每一行输出不超过10个数。

【解答】

所有数的输出连在一起，不便于阅读。

程序修改如下：

```

#include <iostream>

int main()
{
    std::cout << "Enter two numbers:" << std::endl;
    int v1, v2;
    std::cin >> v1 >> v2; // 读入两个数

    // 用较小的数作为下界lower、较大的数作为上界upper
    int lower, upper;
    if (v1 <= v2) {
        lower = v1;
        upper = v2;
    } else {
        lower = v2;
        upper = v1;
    }

    // 输出从lower到upper之间的值
    std::cout << "Values of " << lower << "to "
                << upper << "inclusive are: " << std::endl;
    for (int val = lower, count=1; val <= upper; ++val, ++count) {
        std::cout << val << " ";
        if (count % 10 == 0) //每行输出10个值
            std::cout << std::endl;
    }

    return 0;
}

```

粗黑体部分为主要的修改：用变量count记录已输出的数的个数；若count的值为10的整数倍，则输出一个换行符。

习题1.20

编写程序，求用户指定范围内的数的和，省略设置上界和下界的if测试。假定输入数是7和3，按照这个顺序，预测程序运行结果。然后按照给定的数是7和3运行程序，看结果是否与你预测的相符。如果不相符，反复研究关于for和while循环的讨论直到弄清楚其中的原因。

【解答】

可编写程序如下：

```
// 1-20.cpp
// 省略设置上界和下界的if测试, 求用户指定范围内的数的和

#include <iostream>

int main()
{
    std::cout << "Enter two numbers:" << std::endl;
    int v1, v2;
    std::cin >> v1 >> v2; // 读入数据

    int sum = 0;
    // 求和
    for (int val = v1; val <= v2; ++val)
        sum += val; // sum = sum + val

    std::cout << "Sum of " << v1
              << " to " << v2
              << " inclusive is "
              << sum << std::endl;

    return 0;
}
```

如果输入数据为7和3, 则v1值为7, v2值为3。for语句头中将val的初始值设为7, 第一次测试表达式val <= v2时, 该表达式的值为false, for语句的循环体一次也不执行, 所以求和结果sum为0。

习题1.21

本书配套网站 (http://www.awprofessional.com/cpp_primer) 的第1章的代码目录下有Sales_item.h源文件。复制该文件到你的工作目录。编写程序, 循环遍历一组书的销售交易, 读入每笔交易并将交易写至标准输出。

【解答】

```
#include <iostream>
#include "Sales_item.h"

int main()
{
    Sales_item book;

    // 读入ISBN, 售出书的本数, 销售价格
    std::cout << "Enter transactions:" << std::endl;
    while (std::cin >> book) {
        // 输出ISBN, 售出书的本数, 总收入, 平均价格
        std::cout << "ISBN, number of copies sold, "
                  << "total revenue, and average price are:"
                  << std::endl;
        std::cout << book << std::endl;
    }

    return 0;
}
```

习题1.22

编写程序, 读入两个具有相同ISBN的Sales_item对象并产生它们的和。

【解答】

```

#include <iostream>
#include "Sales_item.h"

int main()
{
    Sales_item trans1, trans2;

    // 读入交易
    std::cout << "Enter two transactions:" << std::endl;
    std::cin >> trans1 >> trans2;

    if (trans1.same_isbn(trans2))
        std::cout << "The total information: " << std::endl
            << "ISBN, number of copies sold, "
            << "total revenue, and average price are:"
            << std::endl << trans1 + trans2;
    else
        std::cout << "The two transactions have different ISBN."
            << std::endl;

    return 0;
}

```

习题1.23

编写程序，读入几个具有相同ISBN的交易，输出所有读入交易的和。

【解答】

```

#include <iostream>
#include "Sales_item.h"

int main()
{
    Sales_item total, trans;

    // 读入交易
    std::cout << "Enter transactions:" << std::endl;

    if (std::cin >> total) {
        while (std::cin >> trans)
            if (total.same_isbn(trans)) // ISBN相同
                total = total + trans;
            else { // ISBN不同
                std::cout << "Different ISBN." << std::endl;
                return -1;
            }
        // 输出交易之和
        std::cout << "The total information: " << std::endl
            << "ISBN, number of copies sold, "
            << "total revenue, and average price are:"
            << std::endl << total;
    }
    else {
        std::cout << "No data?!" << std::endl;
        return -1;
    }

    return 0;
}

```

习题1.24

编写程序，读入几笔不同的交易。对于每笔新读入的交易，要确定它的ISBN是否和以前的交易的ISBN一样，并且记下每一个ISBN的交易的总数。通过给定多笔不同的交易来测试程序。这些交易必须代表多个不同的ISBN，但是每个ISBN的记录应在同一组。

【解答】

```
#include <iostream>
#include "Sales_item.h"

int main()
{
    // 声明变量以保存交易记录以及具有相同ISBN的交易的数目
    Sales_item trans1, trans2;
    int amount;

    // 读入交易
    std::cout << "Enter transactions:" << std::endl;
    std::cin >> trans1;
    amount=1;
    while (std::cin >> trans2)
        if (trans1.same_isbn(trans2))// ISBN相同
            ++amount;
        else {
            // ISBN不同
            std::cout << "Transaction amount of previous ISBN: "
                << amount << std::endl;
            trans1 = trans2;
            amount=1;
        }

    // 输出最后一个ISBN的交易数目
    std::cout << "Transaction amount of the last ISBN: "
        << amount << std::endl;

    return 0;
}
```

习题1.25

使用源自本书配套网站的Sales_item.h头文件，编译并执行1.6节给出的书店程序。

【解答】

可从C++ *Primer* (第4版)的配套网站(http://www.awprofessional.com/cpp_primer)下载头文件Sales_item.h，然后使用该头文件编译并执行1.6节给出的书店程序。

习题1.26

在书店程序中，我们使用了加法操作符而不是复合赋值操作符将trans加到total中，为什么我们不使用复合赋值操作符？

【解答】

因为在1.5.1节中提及的Sales_item对象上的操作中只包含了+和=，没有包含+=操作。（事实上，使用Sales_item.h文件，已经可以用+=操作符取代=和+操作符的复合使用。）

变量和基本类型

习题2.1

int、long和short类型之间有什么差别？

【解答】

它们的最小存储空间不同，分别为16位、32位和16位。一般而言，short类型为半个机器字（word）长，int类型为一个机器字长，而long类型为一个或两个机器字长（在32位机器中，int类型和long类型的字长通常是相同的）。因此，它们的表示范围不同。

习题2.2

unsigned和signed类型有什么差别？

【解答】

前者为无符号类型，只能表示大于或等于0的数。后者为带符号类型，可以表示正数、负数和0。

习题2.3

如果在某机器上short类型占16位，那么可以赋给short类型的最大数是什么？unsigned short类型的最大数又是什么？

【解答】

若在某机器上short类型占16位，那么可以赋给short类型的最大数是 $2^{15}-1$ ，即32767；而unsigned short类型的最大数为 $2^{16}-1$ ，即65535。

习题2.4

当给16位的unsigned short对象赋值100000时，赋的值是什么？

【解答】

34464。

100000超过了16位的unsigned short类型的表示范围，编译器对其二进制表示截取低16位，相当于对65536求余（求模，%），得34464。

习题2.5

float类型和double类型有什么差别？