



全国计算机等级考试命题研究中心 编著
飞思教育产品研发中心
未来教育教学与研究中心 联合监制

全国计算机等级考试

NATIONAL COMPUTER RANK EXAMINATION

上机真题 考点详解



——二级C

上机题库 全面更新

学上机“真题”，以真题为纽带组织全书

攻克难点 突破机考

总结重点、难点，讲解通用解法，简单、实用

智能软件 全真演练

例题、习题上机实践，真考环境下学练“真题”



包含模拟考试
和同步训练



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

全国计算机等级考试

NATIONAL COMPUTER RANK EXAMINATION

上机真题 考点详解

——二级 C

全国计算机等级考试命题研究中心 编著
飞思教育产品研发中心
未来教育教学与研究中心 联合监制

电子工业出版社
Publishing House of Electronics Industry
北京·BEIJING

内容简介

本书依据教育部考试中心最新发布的《全国计算机等级考试考试大纲》，在对最新上机真题进行分析、总结、归纳的基础上编写而成。本书在编写过程中，编者充分考虑等级考试考生的实际特点，并根据考生学习规律进行科学、合理的安排，节省考生的备考时间。

全书共 10 章，主要内容包括：C 程序设计的初步认识，C 语言基本结构，函数，指针，数组，字符串，结构体、共用体和用户定义类型，文件，常见题型及解法，上机真题实战等。

本书配套光盘包括两部分内容：模拟考试和同步训练。通过配套软件的使用，考生可以通过软件对所学知识点进行巩固，真正做到事半功倍。

本书可作为全国计算机等级考试培训和自学用书，尤其适用于上机考试较薄弱的考生。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

全国计算机等级考试上机真题考点详解. 二级 C / 全国计算机等级考试命题研究中心编著.

北京: 电子工业出版社, 2009.11

(飞思考试中心)

ISBN 978-7-121-09574-0

I. 全… II. 全… III. ①电子计算机—水平考试—自学参考资料②C 语言—程序设计—水平考试—自学参考资料
IV. TP3

中国版本图书馆 CIP 数据核字 (2009) 第 170658 号

责任编辑: 杨 鹂 赵树刚

印 刷: 涿州市京南印刷厂

装 订: 涿州市桃园装订有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 889×1194 1/16 印张: 12.5 字数: 560 千字

印 次: 2009 年 11 月第 1 次印刷

印 数: 6 000 册 定价: 29.80 元 (含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前言

计算机等级考试包括笔试和机试两种考查形式，而机试往往又是大多考生比较薄弱的环节。在以往的上机考试中，很多考生采取死记硬背上机试题的方式备考，但由于死记硬背式的记忆容易遗忘，再加上上机考试题库试题的更新，造成很多考生在上机考试中“折戟沉沙”。那么如何才能提高上机考试的过关率呢？

事实上，虽然每年上机考试题库都有不同程度的更新，但考查知识点是固定的，只要学会、学懂上机考试题库中涉及的有限考点，练习数套上机考试真题以熟悉上机考试题型和操作过程，那么通过上机考试便犹如“探囊取物”了。

为帮助考生顺利通过上机考试，全国计算机等级考试专业研究机构——未来教育教学与研究联合飞思教育产品研发中心，在对大量考生和专业等考培训教师跟踪调查的基础上，通过对历年上机真题、最新大纲及命题规律的分析，精心研发了本套图书，旨在帮助考生省时、省力、保险地通过上机考试。

※ 上机考点 全面揭秘

本书中的考点全面覆盖历年真题题库，系统讲解了上机考试中所有的考点和题型。上机考试有别于笔试，但市场上的教程或辅导类等考图书大都是针对笔试的，造成很多考生学完之后还是对上机考试一窍不通。通过本书的使用，能有效弥补上机考试的“短板”。

※ 上机真题 临摹结合

本书中的所有例题和习题都是历年真题或真题的一部分，在学习的同时记忆了上机题库试题。设置相应章节对上机试题中的常见题型进行分析，同时传授解题方法，使考生轻松掌握上机真题的解法。此外，书中所有例题、习题所对应的源程序文件在光盘上都能找到，考生可以边学边练。

※ 智能软件、真题实战

随书附赠的软件包含两部分内容：同步训练和模拟考试。同步训练模块将书中的所有例题和习题进行再现，方便考生使用，并且其中的真题演练功能还可以对考生的做题结果进行评分；模拟考试模块力求从登录、抽题、答题、交卷等各方面与真实考试保持一致，评分原理也与真实考试一模一样，其中试题采用了上机考试真题，且提供详细的试题答案和视频解析。通过配套软件的使用，考生可以对所学知识点进行巩固，真正做到事半功倍。

“人之学问知能成就，犹骨象玉石切磋琢磨也”。大量考生备考实例表明：好的工具+好的学习方法+努力=成功，合理使用好本书及智能考试模拟软件，加上适当的练习，就能轻松顺利地通过上机考试。

编委会

联系方式

咨询电话：(010) 88254160 88254161-67

电子邮件：support@fecit.com.cn

服务网址：<http://www.fecit.com.cn> <http://www.fecit.net>

通用网址：计算机图书、飞思、飞思教育、飞思科技、FECIT

丛书编委会

丛书主编 詹可军

学科主编 侯俊伯

编 委 (排名不分先后)

丁海艳	万克星	马立娟	亢艳芳
王 伟	王 亮	王强国	王 磊
卢文毅	卢继军	任海艳	伍金凤
刘之夫	刘金丽	刘春波	孙小稚
张 迪	张仪凡	张海刚	李 静
李明辉	李志红	杨 力	杨 闯
杨生喜	花 英	陈秋彤	周 辉
孟祥勇	欧海升	武 杰	范海双
郑 新	姜 涛	姜文宾	胡 杨
胡天星	赵 亮	赵东红	赵艳平
侯俊伯	倪海宇	高志军	高雪轩
董国明	谢公义	韩峻余	熊化武

目 录

第1章 C 程序设计的初步认识	(1)
考点1 C 语言程序设计的格式	(2)
考点2 数据类型	(3)
考点3 算术表达式	(5)
考点4 赋值表达式	(7)
考点5 自加、自减运算符	(9)
第2章 C 语言基本结构	(11)
考点1 格式输入与输出	(12)
考点2 if 语句构成的选择结构	(15)
考点3 逻辑运算符与表达式	(17)
考点4 switch 语句及用 switch 语句和 break 语句构成的选择结构	(19)
考点5 while 语句和用 while 语句构成的循环结构	(21)
考点6 do...while 语句和用 do...while 语句构成的循环结构	(22)
考点7 for 语句和用 for 语句构成的循环结构	(24)
考点8 循环的嵌套	(25)
考点9 选择结构与顺序结构、循环结构的混合使用	(26)
第3章 函数	(31)
考点1 函数的定义	(32)
考点2 函数参数及返回值	(34)
考点3 函数的调用	(35)
考点4 迭代算法和递归算法	(36)
第4章 指针	(41)
考点1 指针变量	(42)
考点2 指针变量的操作	(43)
考点3 函数之间的地址传递	(44)
第5章 数组	(47)
考点1 一维数组的定义和一维数组元素的引用	(48)
考点2 排序算法	(50)
考点3 一维数组和指针	(52)
考点4 函数之间对一维数组的引用	(54)
考点5 二维数组的定义和二维数组元素的引用	(56)
考点6 二维数组和指针	(58)

第6章 字符串	(63)
考点1 字符串与字符数组	(64)
考点2 字符串指针	(67)
考点3 字符串处理函数	(69)
第7章 结构体、共用体和用户定义类型	(75)
考点1 结构体变量的定义与表示方法	(76)
考点2 指向结构体变量的指针及链表的操作	(79)
第8章 文件	(95)
考点1 文件的打开与关闭	(96)
考点2 文件的读/写	(99)
考点3 文件检测函数	(101)
第9章 常见题型及解法	(105)
考点1 数学类问题	(106)
考点2 字符串类问题	(112)
考点3 数组类问题	(121)
考点4 结构体类问题	(127)
第10章 上机真题实战	(143)
10.1 上机考试环境及流程	(144)
10.2 上机考试真题	(146)
10.3 参考答案及解析	(175)
附录 A 最新大纲专家解读	(191)

第1章

C 程序设计的初步认识

• 本章导读 •

本章是 C 语言必考知识,在填空题、改错题和编程题中都有考查。C 语言语法简洁、紧凑,使用方便、灵活,具有丰富的数据类型和预定义运算符,同时自定义数据类型也十分灵活。在 C 语言中,基本的算术运算符是: +、-、*、/、%, 分别是加、减、乘、除和求余运算符,赋值号“=”称为赋值运算符。除此之外,C 语言中还包括很多其他的运算符。

本章难点:算术表达式及赋值表达式的相关知识,需要考生重点学习,以便熟练掌握。

• 本章考点 •

序号	考点	重点掌握	难度	抽中几率		
				填空题	改错题	编程题
1	C 语言程序设计的格式	注意 C 语言程序的编写格式	★		6%	
2	数据类型	常量、变量、数据类型	★★★★★	18%	23%	81%
3	算术表达式	算术表达式及运算符的优先级和结合性	★★★★★	11%	7%	7%
4	赋值表达式	复合赋值运算符、类型转换	★★★★★	18%	11%	15%
5	自加、自减运算符	自加、自减运算符	★★★★★	13%	13%	22%



考点1 C 语言程序设计的格式

C 程序的结构特点如下。

(1) C 程序是由函数构成的, 一个 C 源程序至少且仅包含一个 main 函数, 即主函数。程序的执行从主函数也就是 main 函数开始。

(2) 函数体必须用 {} 括起来。

(3) 每个执行语句都必须以分号 (;) 结尾, 预处理命令, 函数头和花括号 {} 之后不加分号。

例如:

```
#include <stdio.h>                /* 预处理命令后不加分号 */
void main()                        /* 函数头后不加分号 */
{
    printf("Hello World!");        /* 执行语句以分号结束 */
}                                  /* 函数体 {} 后不加分号 */
```

【例 1.1】如下程序的功能是求圆形面积, 请改正程序中的错误。

程序如下:

```
#include <stdio.h>
main()
{ double r, area;
/* * * * * * found * * * * * */
    r=5                                /* 将圆形半径的值赋给变量 r */
    area=3.14*r*r;                    /* 计算圆形面积并存储到变量 area 中 */
    printf("area=%f\n", area);        /* 输出圆形的面积 */
}
```

【分析】该程序由一个主函数组成, 程序首先定义了两个变量 r 和 area, 用以存放圆形的半径和面积, 然后给半径赋值为 5, 接着计算圆形的面积, 最后将圆形的面积输出到终端。观察给定程序内容, r=5 为一条赋值语句, 没有以 ";" 结束, 故需要在该语句的后面加上 ";"。

小提示

1. 上机考试时, 每个改错题要考查 2~4 个错误, 且每个错误均出现在一条 /* * * * * * found * * * * */ 语句的后面。
2. 做改错题前先编译一下程序是个比较好的习惯, 因为系统会帮我们找到一些语法性的错误。

考点分析

C 语言程序的基本格式每年的考题都会涉及, 比较简单。主要考查形式有:

- 语句缺少 ";"。
- "{" 或 "(" 的匹配。
- 运算符或关键字的书写错误。

3) 字符串常量

字符串常量是由一对双引号括起的字符序列,如"hello"、"12345"等都是合法的字符串常量。

注意,字符串常量和字符常量是不同的量,它们之间的区别如下。

- 字符常量由单引号括起来,字符串常量由双引号括起来。
- 字符常量只能是单个字符,字符串常量则可以含一个或多个字符。
- 可以把一个字符常量赋予一个字符变量,但不能把一个字符串常量赋予一个字符变量。
- 字符常量占一个字节的内存空间,字符串常量占的内存字节数等于字符串中字节数加1,最后一个字节中存放字符'\0' (ASCII 码为0),即字符串结束标志。

例如:字符串"hello"在内存中所占的字节为:

h	e	l	l	o	\0
---	---	---	---	---	----

4. 变量的初始化

前面我们讲了整型变量、实型变量和字符型变量,系统在定义这些变量的时候,只是为它们分配了内存单元,并没有给它们赋值。在编程时,引用一个没有赋值的变量是没有意义的,因此需要对变量初始化。

1) 变量初始化的两种方法

①在定义变量的时候初始化。

例如: `int a=1;`

`float f=3.14;`

也允许对被定义的变量的一部分赋初值。

例如: `char a,b,c='E';` /* 同时声明3个变量a、b、c,但只对c赋初值为E */

小提示

注意:如果对几个变量赋予同一个值,应写成:

`int a=3,b=3,c=3;`

不是:

`int a=b=c=3;` /* 同时声明3个变量a、b、c,但只对c赋初值为3 */

②先定义变量,用到的时候再初始化。

例如: `int b;`

`b=100;`

小提示

在考试中我们可能会因为输入的错误而导致程序的错误,这时我们需要学会使用VC++的调试功能,能看懂VC++的信息提示。

例如:如果在声明部分有语句:

`int student;` /* 定义或声明了一个整型变量student */

而在后边的执行语句中不小心将student错写成stadent,如:

`stadent=30;`

在编译的时候将检查出stadent未定义,输出“Undefined symbol stadent in function main”的信息,提示用户检查错误。

2) 常见的初始化操作

①循环变量初始化。在循环结构中,我们常使用循环变量来控制循环的次数,此时循环变量被初始化为0、1或某一个特定值。

例如: `int i;`

```
for(i=1;i<=10;i++)
{ ... }
```

②变量用于存储累加或累乘后的结果。当使用循环语句对变量进行累加或累乘的操作时,需要对该变量赋一个初值。

例如:求 $1 \times 2 \times 3 \times \cdots \times 10$ 的值。

```
int i, sum;                /* i 为循环变量, sum 存储计算结果 */
sum = 1;                   /* 给 sum 赋初值 1 */
for(i=1;i<=10;i++) sum = sum * i; /* 循环 10 次,每次都使 sum 乘以增量 i */
```

小提示

习惯上,如果变量用于累加操作,将其初始化为0;如果变量用于累乘操作,将其初始化为1。

【例 1.2】如下程序的功能是:求公式 $s = 1 + 2 + 3 + \cdots + n$ 的前 100 项的值。请填写程序中的空白。

程序如下:

```
#include <stdio.h>
void main()
{ int i,s;                /* 定义循环变量 i 和累加变量 s */
  s = 1;                /* 给累加变量赋初值 */
  for(i = 2; i <= 100; i++) /* 循环 100 次 */
    s = s + i;             /* 累加 */
  printf("S = %d\n", s);   /* 输出计算结果 */
}
```

【分析】该程序由一个主函数组成,定义了两个整型变量 i 和 s ,并为 s 赋初值,然后使用循环语句累加 s 的值,最后将计算结果输出。

空白 1:变量 s 用于存放累加的和,所以我们将其初始化为 0。

空白 2:使用循环变量 i 控制循环的次数,程序要执行 100 次循环,每执行一次循环,变量 i 的值加 1,当 i 的值大于 100 时循环结束,因此变量 i 初始化为 1。



考点 3 算术表达式

1. 算术运算符

(1)加法运算符“+”:双目运算符,即应有两个量参与加法运算,如 $x + y$ 、 $1 + 5$ 等,具有右结合性。

(2)减法运算符“-”:双目运算符。但“-”也可作为负值运算符,此时为单目运算符,如 $-a$ 、 -15 等,具有左结合性。

(3)乘法运算符“*”:双目运算符,具有左结合性。

(4)除法运算符“/”:双目运算符,具有左结合性。参与运算量均为整型时,结果也为整型,舍去小数。

如果运算量中有一个是实型,则结果为双精度实型。

(5)求余运算符(模运算符)“%”:双目运算符,具有左结合性。要求参与运算的量均为整型。求余运算的结果是两数相除后的余数。

2. 算术表达式

算术表达式是由算术运算符和括号连接起来的式子。

例如: $a + b$

$(x * 2) / y$

$(a - b) / 5 + (m + n) * 3$

$\sin(x) + \sin(y)$

3. 运算符的优先级、结合性

1) 算术运算符的优先级

算术运算符和圆括号的优先级由高到低次序如下所示:

() (圆括号)、+ (求正)、- (求负)、* (乘)、/ (除)、% (求余)、+ (加)、- (减)

2) 算术运算符的结合性

算术运算符和圆括号的结合性:在以上所列运算符中,只有单目运算符“+”和“-”的结合性是从右到左的,其余运算符的结合性都是从左到右。

例如: $(5 + 3) / 2$ 的运算结果是 4

$5 + 3 / 2$ 的结果是 6

$5 * -2$ 的结果是 -10

【例 1.3】如下程序的功能是:将 a、b 中的两个两位正整数合成一个新的整数放在 c 中。合并的方式是:将 a 中的十位和个位数依次放在 c 的百位和个位上,b 中的十位和个位数依次放在 c 的十位和千位上。如 a = 45, b = 12, 调整后 c = 2415。请填写程序中的空白。

程序如下:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int a=45,b=12;           /* 定义两个整型变量 a、b 并赋值 */
```

```
    long c;                  /* 定义变量 c 存放新整数 */
```

```
    c = _____;          /* 计算新整数 c */
```

```
    printf("The result is :%ld\n",c);      /* 输出 c */
```

```
}
```

【分析】要解决本题,首先要得到两个整数 a 和 b 的各个位上的数值,然后将它们组合成新数 c。

我们可以通过将一个两位数对 10 做除法或求余的方法,来得到这个数的十位和个位数值,即: $45 / 10 = 4$ (a 的十位), $45 \% 10 = 5$ (a 的个位), $12 / 10 = 1$ (b 的十位), $12 \% 10 = 2$ (b 的个位)。因此,空白处应为: $(b \% 10) * 1000 + (a / 10) * 100 + (b / 10) * 10 + (a \% 10)$ 。

考点分析

算术运算符和算术表达式是常考查的知识,既可能单独考查,也可能结合题目考查。主要考查形式有:

- 除法运算符“/”和求余运算符“%”的区别和应用。

- 根据运算符的优先级与结合性确定算术表达式。

小提示

我们可以用以下公式来求得多位数 S 的第 n 位 (n 不大于 S 的位数) 上的数值 k :

$$k = S / 10^{(n-1)} \% 10$$

考点4 赋值表达式**1. 赋值运算符和赋值表达式**

在 C 语言中,赋值运算符为“=”。由“=”连接的式子称为赋值表达式。

【格式】

变量名 = 表达式

例如: $x = a + b$

$w = \sin(a) + \sin(b)$

赋值表达式的功能是计算表达式的值再赋予左边的变量。赋值运算符具有右结合性。

因此 $a = b = c = 5$ 可理解为 $a = (b = (c = 5))$

在其他高级语言中,赋值构成了一个语句,称为赋值语句。而在 C 语言中,把“=”定义为运算符,从而组成赋值表达式。凡是表达式可以出现的地方均可出现赋值表达式。

例如:

$x = (a = 5) + (b = 8)$ 是合法的。它的意义是把 5 赋给变量 a , 8 赋给变量 b , 再把 a 与 b 相加, 和赋给变量 x , 因此最后变量 x 应为 13。

在 C 语言中也可以组成赋值语句, 按照 C 语言的规定, 任何表达式在其末尾加上分号即可构成语句。因此如 $x = 8; a = b = c = 5;$ 都是赋值语句。

2. 复合赋值表达式

在赋值运算符之前加上其他运算符可以构成复合赋值运算符。其中与算术运算有关的复合赋值运算符有: $+=$ 、 $-=$ 、 $*=$ 、 $/=$ 、 $\%=$ 。复合赋值运算符的优先级与赋值运算符的优先级相同。

例如:

表达式 $n += 1$ 的运算规则等价于 $n = n + 1$ 。

【考查形式】赋值表达式在二级考试中是必考的, 因为任何变量都需要赋值后才能进行其他操作。其考查形式如下。

- 普通变量赋值时。
- 将表达式的值赋值给一个变量时。
- 运用复合表达式进行计算时, 这时要用到复合赋值运算符: $+=$ 、 $-=$ 、 $*=$ 、 $/=$ 、 $\%=$ 。

【例 1.4】分析下列给定程序, 输出表达式的结果。

程序如下:

```
#include <stdio.h>
main()
```

考点分析

尽管二级考试中考查的不是很多但要注意, 主要考查形式如下:

- 变量赋值。
- 使用符合赋值表达式完成累加或累乘操作。
- 两个整数做除法时进行类型转换。

```

{  int a,b,c,d,e;    /* 定义5个整型变量a、b、c、d、e */
    a=40; b=30; c=15; d=60; e=20;    /* 变量赋值 */
    a+=5;                /* 等价于 a=a+5; */
    b-=a;                /* 等价于 b=b-a; */
    c*=10;               /* 等价于 c=c*10; */
    d/=e;                /* 等价于 d=d/e; */
    e%=15;               /* 等价于 e=e%15; */
    printf("a=%d, \nb=%d, \nc=%d, \nd=%d, \ne=%d\n", a,b,c,d,e);
}

```

【分析】该程序中,首先给变量a、b、c、d、e分别赋初值,然后通过复合赋值运算分别给上述5个变量重新赋值。

a+=5 等价于 a=a+5=40+5=45;

b-=a 等价于 b=b-a=30-45=-15;

c*=10 等价于 c=c*10=15*10=150;

d/=e 等价于 d=d/e=60/20=3;

e%=15 等价于 e=e%15=20%15=5;

因此,最后的输出结果为:a=45,b=-15,c=150,d=3,e=5。

小提示

赋值号“=”称为赋值运算符,“==”是C语言中的关系运算符“等于”,此处应重点区分。如a=0,赋值语句,即给变量a赋值为0;如a==0,即需判断条件表达式a==0的真假。

3. 类型转换

1) 强制类型转换

强制类型转换是通过类型转换运算来实现的。

【格式】

(类型说明符)(表达式)

【说明】

上述形式中,(类型说明符)称为强制类型转换运算符,利用该运算符可以将一个表达式的值转换成指定的类型。

例如:(double)(10%3); /* 将10%3的结果1转换成双精度型 */

(float)a; /* 将变量a转换为实型 */

(int)(x+y); /* 将x+y的结果转换为整型 */

2) 在算术运算符中实现

在C语言中进行算术运算时,系统可以进行自动的强制类型转换。

例如:在C语言的除法运算中,如果运算量中有一个是实型,则结果默认转换为双精度实型,这与强制类型转换等价,如1.0/n等价于(double)(1/n)(将1/n的结果转换为双精度型)。

考点5 自加、自减运算符

1. 自加运算符

自加1运算符记为“++”，其功能是使变量的值自增1。

自加运算符形式如下。

- ++i: i 自加1后再参与其他运算。
- i++: i 参与运算后, i 的值再自加1。

2. 自减运算符

自减1运算符记为“--”，其功能是使变量的值自减1。

自减运算符形式如下。

- --i: i 自减1后再参与其他运算。
- i--: i 参与运算后, i 的值再自减1。

【注意】自加1、自减1运算符均为单目运算符,具有右结合性。在以上4种形式中,理解和使用上容易出错的是i++和i--。特别是当它们出在较复杂的表达式或语句中时,常常难于弄清,因此应仔细分析。

【例1.5】输出以下程序结果。

程序如下:

```
#include <stdio.h>
main()
{
    int i=8;                /* 整型变量 i 赋初值为 8 */
    printf("%d\n", ++i);    /* 输出 ++i */
    printf("%d\n", --i);    /* 输出 --i */
    printf("%d\n", i++);    /* 输出 i++ */
    printf("%d\n", i--);    /* 输出 i-- */
    printf("%d\n", -i++);   /* 输出 -i++ */
    printf("%d\n", -i--);   /* 输出 -i-- */
}
```

【分析】由程序可知, i 的初值为8, ++i 即 i 加1后输出, 故输出和 i 均为9; --i 即 i 减1后输出, 故输出和 i 均为8; i++ 即输出 i 为8之后再加1, 故 i 为9; i-- 即输出 i 为9之后再减1, 故 i 为8; -i++ 即输出 -i 为-8之后 i 再加1, 故 i 为9, -i-- 即输出 -9之后再减1, 故 i 最后为8。输出的顺序依次为: 9 8 8 9 -8 -9。

小提示

重点区别 i++ 和 ++i, i-- 和 --i。

真题演练

1. 请编写函数, 其功能是计算并输出表达式 $s = 1 + 2 + 3 + \dots + 50$ 的值。

考点分析

自加自减运算符在二级考试中常考, 主要考查形式如下:

- 题目中要求的自加、自减运算, 也可用算术运算符实现。
- 循环语句中循环变量增值时的使用, 这时要注意这4种形式的不同。

2. 输出下列程序结果。

```
#include <stdio.h>
main()
{ int i=5,j=5,p,q;
  + + i;
  j + +;
```

```
}
    p=(i++)+j;
    q=(++j)+i;
    printf("%d,%d,%d,%d",p,q,i,j);
}
```

3. 输出下列程序结果。

```
#include <stdio.h>
main()
{ float f=5.75;
  printf("f=%d\n",(int)f);
}
```

4. 输出下列程序结果。

```
#include <stdio.h>
main()
{ long x,y;
  int a,b,c,d;
  x=5;
  y=6;
  a=7;
```

```
}
    b=8;
    c=x+a;
    d=y+b;
    printf("c=x+a=%d,d=y+b\n",c,d);
}
```

5. 请指出以下程序内容的错误。

```
#include <stdio.h>
main()
/* * * * * found * * * * */

float r,s; /* r 为半径,s 为圆面积 */
```

```
}
    r=5.0;
    s=3.14159*r*r;
/* * * * * found * * * * */
    printf("%f\n",s)
}
```