

高等学校21世纪教材

GAODENG XUEXIAO 21 SHIJI JIAOCAI

C语言 程序设计 题解与实验指导 (修订本)

● 徐士良 编著

COMPUTER

人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目 (CIP) 数据

C 语言程序设计题解与实验指导/徐士良编著. —修订本. —北京:人民邮电出版社, 2003.7
高等学校 21 世纪教材
ISBN 7-115-11322-X

I. C... II. 徐... III. C 语言—程序设计—高等学校—教学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2003) 第 043866 号

内 容 提 要

本书是高等学校 21 世纪教材《C 语言程序设计教程》(修订本)的配套用书。本书的第一部分给出《C 语言程序设计教程》(修订本)一书中所有习题的参考答案,其中所有的程序都经过调试;第二部分介绍 C 语言的实验环境,由浅入深地安排大量的实验,并简要介绍程序的调试。

本书不仅可以与《C 语言程序设计教程》(修订本)配套使用,也可以作为其他 C 语言程序设计教材的习题集与实验指导。本书可作为本科大专院校各专业的学生学习 C 语言程序设计的参考书,也可供计算机培训班或其他自学者使用。

高等学校 21 世纪教材

C 语言程序设计题解与实验指导 (修订本)

- ◆ 编 著 徐士良
责任编辑 滑 玉
- ◆ 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>
读者热线 010-67129260
北京汉魂图文设计有限公司制作
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 9.5
字数: 227 千字 2003 年 7 月第 2 版
印数: 13 001-18 000 册 2003 年 7 月北京第 4 次印刷

ISBN 7-115-11322-X/TP · 3482

定价: 14.00 元

本书如有印装质量问题,请与本社联系 电话:(010)67129223

丛书前言

当今世界，科学技术突飞猛进，知识经济已见端倪，国际竞争日趋激烈。教育在综合国力的形成中处于基础地位，国力的强弱将越来越取决于劳动者的素质，取决于各类人才的质量和数量，这对于培养和造就我国 21 世纪的一代新人提出了更加迫切的要求。21 世纪初，我国高等教育呈快速发展的势头。教材是体现教学内容和教学方法的知识载体，是进行教学的基本工具，也是深化教育教学改革、全面推进素质教育、培养创新人才的重要保证。因此，高等教育教材建设必须有一个与之相适应的快速发展。

随着计算机软硬件的不断升级换代，计算机教学内容也随之更新，尤其随着教育部“高等教育面向 21 世纪教育内容与课程体系改革”计划的实施，对教材也提出了新的要求。为此我们聘请了国内高校计算机教学方面知名的专家教授，精心策划编写了这套“高等学校 21 世纪教材”。

为真正实施精品战略，组织编写好这套教材，我们在国内高校做了系统、详细的调查，对教育部制订的教育计划做了认真的研究，还对国内外已出版的教材做了理性的分析，确立了依托国家教育计划、传播选进教学理念、为培养符合社会需要的高素质创新型人才服务的宗旨。

在本套教材的策划过程中，我们多次组织了由专家及高校一线教师参加的研讨会，对现有比较出色的教材的特点及优点进行了分析，博采众长，力求实现教材权威性与实用性的完美结合。

本套教材有如下特点：

1. 考虑到全国普通高等院校学生的知识、能力、素质的特点和实际教学情况，在编写教材时把重点放在基本理论、基础知识、基本技能与方法上。
2. 紧密结合当前技术的新发展，在阐述理论知识的同时侧重实用性。
3. 力求在概念和原理的讲述上严格、准确、精练，理论适中，实例丰富，写作风格上深入浅出，图文并茂，便于学生学习。
4. 为适应当前高校课程种类多、课时数要压缩的教学特点，教材不仅篇幅有很大的压缩，而且均配有电子教案，以满足现代教学新特点的需要，做到易教易学。
5. 所选作者均是国内有丰富教学实践经验的知名专家、教授，所编教材具有较高的权威性。

教育的改革将不会停止，教材也将会不断推陈出新。目前本套教材即将推出，将接受广大教学第一线教师的检验。

由于我们的水平和经验有限，这批教材在编审、出版工作中还存在不少缺点和不足，希望使用本套教材的学校师生和广大读者提出批评和建议，以便改进我们的工作，使教材质量不断提高。

编者的话

(修订本)

本书是与《C语言程序设计教程》(修订本)一书配套的教材,也能作为其他C语言程序设计教材的习题集与实验教材。本书主要包括两大部分:第一部分给出了《C语言程序设计教程》(修订本)一书中所有习题的解题思路与参考答案,其中的所有程序都经过调试并运行通过;第二部分介绍了两种常用文本编辑的使用,C编译环境以及程序的调试,并为《C语言程序设计教程》(修订本)一书中的各章安排了实验内容,最后还安排了综合性的实验内容。

在本书中,对每个实验都明确规定了实验目的,提出了具体要求,还给出了方法说明。

在本书的附录中,给出了Turbo C编译错误信息,供读者在调试程序时查阅。由于作者水平有限,书中难免存在错误和不妥之处,恳请读者批评指正。

作者

2003年5月

目 录

第一部分 《C程序设计教程》（修订本）习题解答	1
第1章 绪论	1
第2章 程序的基本组成	3
第3章 选择结构	14
第4章 循环结构	23
第5章 模块设计	32
第6章 数组	40
第7章 指针	50
第8章 结构体与联合体	59
第9章 文件	69
第10章 位运算	73
第二部分 实验指导	75
第11章 文本编辑	75
11.1 屏幕编辑程序 ED	75
11.2 文本编辑程序 EDIT	77
第12章 Turbo C 编译环境	82
12.1 Turbo C 命令行编译方式	82
12.2 Turbo C 集成编译环境	83
12.3 实例	91
第13章 程序的调试	97
13.1 程序设计的风格	97
13.2 程序的调试	98
第14章 上机实验内容	101
14.1 C程序的运行过程	101
14.2 简单程序设计	103
14.3 选择结构程序设计	105
14.4 循环结构程序设计	107
14.5 模块化程序设计	109
14.6 使用数组的程序设计	111

14.7 使用指针的程序设计	114
14.8 使用结构体的程序设计	117
14.9 使用文件的程序设计	120
14.10 位运算	123
14.11 综合程序设计	124
附录 Turbo C 编译错误信息	131
参考文献	144

第一部分 《C 程序设计教程》 (修订本) 习题解答

第 1 章 绪 论

1.01 程序设计语言与算法描述语言有什么本质的区别？

答：用算法描述语言所描述的流程，一般不能直接作为程序来执行，最后还需转换成用某种程序设计语言所描述的程序。算法描述语言与程序设计语言最大的区别就在于，算法描述语言比较自由，不像程序设计语言那样受语法的约束，只要描述得人们能理解就行，而不必考虑计算机处理时所要遵循的规定或其他一些细节。

1.02 有人说：“程序设计就是编制程序。”这句话对不对？为什么？

答：这是不对的，至少是不全面的。实际上，程序设计包括多方面的内容，而具体编制程序只是其中的一个方面。有人将程序设计描述成如下的一个公式：

程序设计 = 算法 + 数据结构 + 方法 + 工具

由此可以看出，在整个程序设计的过程中，要涉及到算法的设计、数据结构的设计、方法的设计和工具的选择等诸多方面。

1.03 结构化程序设计有哪些特点？

答：结构化程序设计要求把程序的结构限制为顺序、选择和循环三种基本结构，以便提高程序的可读性。这种结构化程序具有以下两个特点：

(1) 以控制结构为单位，只有一个入口和一个出口，使各单位之间的接口比较简单，每个单位也容易被人们所理解。

(2) 缩小了程序的静态结构与动态执行之间的差异，使人们能方便、正确地理解程序的功能。

1.04 在模块化设计中，划分程序模块的原则是什么？按功能划分模块有什么优点？

答：在模块化设计中，划分程序模块的原则如下。

1. 按功能划分模块

划分模块的基本原则是使每个模块都易于理解。按照人类思维的特点,按功能来划分模块最为自然。在按功能划分模块时,要求各模块的功能尽量单一,各模块之间的联系尽量少。满足这些要求的模块有以下几个优点:

- (1) 其可读性和可理解性都比较好;
- (2) 各模块间的接口关系比较简单;
- (3) 当要修改某一功能时,只涉及到一个模块;
- (4) 其他应用程序可以充分利用已有的一些模块。

2. 按层次组织模块

在按层次组织模块时,一般上层模块只指出“做什么”,只有在最底层的模块中才精确地描述“怎么做”。

1.05 设分段函数如下:

$$y = \begin{cases} x^2 + 1 & x \geq 0 \\ x^2 - 1 & x < 0 \end{cases}$$

分别使用自然语言、算法描述语言、结构化流程图,描述当输入一个 x 值后计算并输出该函数值的处理流程。

解:用自然语言描述如下:

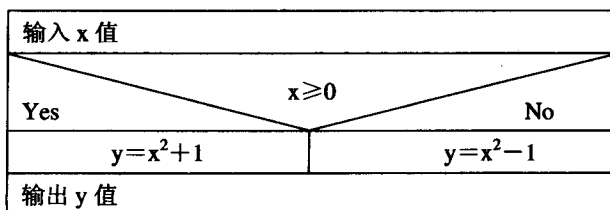
- (1) 输入 x 值。
- (2) 判断 x 值:
 - 若 $x \geq 0$, 则计算 $y = x^2 + 1$;
 - 若 $x < 0$, 则计算 $y = x^2 - 1$ 。
- (3) 输出 y 值。
- (4) 结束

用算法描述语言描述如下:

```

INPUT  x
IF x ≥ 0 THEN y = x2 + 1
ELSE y = x2 - 1
OUTPUT y
END
  
```

用结构化流程图描述如下:



1.06 设多项式如下:

$$s = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \dots + \frac{(-1)^n}{2n+1}x^{2n+1}$$

分别使用自然语言、算法描述语言、结构化流程图,描述当输入 x 值后计算并输出该多项式值的处理流程,直到最后一项的绝对值小于 0.000001 为止。

解:用自然语言描述如下:

- (1) 输入 x 值
- (2) $s=x; n=0; f=x$
- (3) $f=-f; n=n+1; f=f \times x \times x; d=f/(2n+1); s=s+d$
- (4) 若 $|d| \geq 0.000001$, 则转 (3)
- (5) 输出 s
- (6) 结束

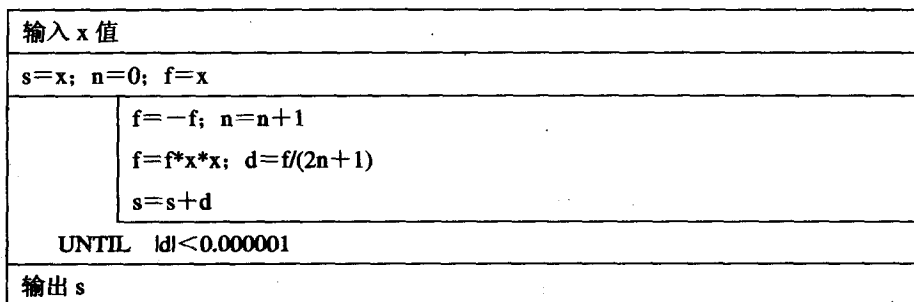
用算法描述语言描述如下:

```

INPUT x
s=x; n=0; f=x
Loop: f=-f; n=n+1; f=f*x*x; d=f/(2n+1); s=s+d
IF |d| ≥ 0.000001 THEN GOTO Loop
OUTPUT s
END

```

用结构化流程图描述如下:



1.07 试举例说明自顶向下、逐步细化的设计过程。

答: (略)

第2章 程序的基本组成

2.01 在下列 C 程序的前两行中填入应包含的文件名:

```

#include "      "
#include "      "

```

```

main()
{ float a, b, z;
  printf("input a and b :");
  scanf("%f, %f", &a, &b);
  z=exp(a*a+b*b);
  printf("z=%f\n", z);
}

```

解：该程序中用到了输入函数 `scanf()` 和输出函数 `printf()`，因此，应包括输入输出函数库头文件 `stdio.h`；该函数中还用到数学库中的指数函数 `exp()`，因此，应包括数学库头文件 `math.h`。综上所述，填入的文件名分别为：`stdio.h` 与 `math.h`。

2.02 下列标识符中，哪些是合法的 C 变量名：

a_qwe	zx-123	\$a234	_a_sdf	qw.c	x/y
a*b	%jkh	xy%c	_1234	1234_	12_34
c1_2	x4_5_6	new_r	root_1	_root1	12345

解：在 C 语言中，变量名是以字母或下划线开头，后跟若干个字母、数字字符或下划线。因此，合法的变量名为

a_qwe _a_sdf _1234 c1_2 x4_5_6 new_r root_1 _root1

2.03 设有下列定义和输入语句：

```

int x, y;
char c, d;
scanf("%d%d", &x, &y);
scanf("%c%c", &c, &d);

```

如果要求变量 `x`, `y`, `c`, `d` 的值分别为 20, 30, X, Y，则正确的数据输入格式是什么？

解：C 语言规定，整型格式说明符中若没有宽度说明，则两输入数据间用空格，或 Tab 键，或回车键分隔；在用 `%c` 作为字符型格式说明符时，各输入的字符之间不用分隔。

对于第一个格式输入语句应输入的整型数据为“20<空格>30”。此时，不能按<回车>键，因为如果按了<回车>键，该<回车>符就被第二个格式输入语句中的变量字符型 `c` 接收，达不到为字符型变量 `c` 赋字符 `X` 的要求。

对于第二个格式输入语句应输入字符 `X` 与 `Y`（具体输入时不必输入其界限符单撇号）。其中第一个字符 `X` 既作为第一个输入语句中输入数值型数据的结束符，又是在第二个输入语句中为字符型变量 `c` 提供的字符。字符 `Y` 是在第二个输入语句中为字符型变量 `d` 提供的字符。

因此，正确的数据输入格式为

20<空格>30XY<回车>

2.04 设有下列 C 程序：

```

#include "stdio.h"
main()

```

```

{ int x=4617;
  printf("x=%8d\n", x);
  printf("x=%-8d\n", x);
}

```

这个程序运行的结果是什么？

解：C语言规定：若格式说明符%后没有“-”号，则在输出数据时为右对齐，左边补空格；但若格式说明符%后有“-”号，则在输出数据时为左对齐，右边补空格。

因此，本程序的运行结果如下：

```

x=    4617 (前面有4个空格)
x=4617    (后面有4个空格)

```

2.05 设有输入语句如下：

```
scanf("x=%d, y=%d, z=%d", &x, &y, &z);
```

为使变量 x 的值为 12，变量 y 的值为 34，变量 z 的值为 62，则从键盘输入数据的正确格式是什么？

解：C语言规定，输入格式说明符中的其他字符应原样输入。因此，在本语句格式控制中的字符“x=”，“， y=”与“， z=”也需要输入。

因此，从键盘输入数据的正确格式如下：

```
x=12, y=34, z=62<回车>
```

2.06 试写出下列各项的 C 语言表达式：

(1) a 与 b 之一为零，但不能同时为零。

(2) $10 < x < 100$ 或 $x < 0$ 但 $x \neq -2.0$ 。

(3) $\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$

(4) 圆心在原点，大圆半径为 r_2 、小圆半径为 r_1 的圆环（包括两个圆周）。

(5) m 能被 5 或 7 整除，但不能同时被它们整除。

(6) $\frac{1}{2}(x \times y + \frac{x+y}{4a})$

解：(1) $(a*b==0)\&\&(a+b!=0)$ 或

$(a==0)\&\&(b!=0)\|\!(a!=0)\&\&(b==0)$

(2) $(x > 10.0)\&\&(x < 100.0)\|\!(x < 0.0)\&\&(x != -2.0)$

(3) $\exp(-x*x/2.0)/\text{sqrt}(2.0*3.1415926)$

(4) 用极坐标表示为 $(r >= r1)\&\&(r <= r2)$ 或

用直角坐标表示为 $(x*x + y*y >= r1*r1)\&\&(x*x + y*y <= r2*r2)$

(5) $((m\%5==0)\|\!(m\%7==0))\&\&(m\%35!=0)$ 或

$(m\%5==0)\&\&(m\%7!=0)\|\!(m\%5!=0)\&\&(m\%7==0)$

(6) $(x*y + (x+y)/(4*a))/2.0$

2.07 设 $a=3, b=4, c=5$ 。试写出下列 C 语言表达式的值:

- (1) $a+b>c \ \&\& \ b==c$
- (2) $a \ \&\& \ b+c \ \parallel \ b-c$
- (3) $!(a+b)+c-1 \ \&\& \ b+c/2$
- (4) $a \ \parallel \ b+c \ \&\& \ b-c$
- (5) $!(x==a) \ \&\& \ (y==b) \ \&\& \ 0$

解: (1) 0 (2) 1 (3) 1 (4) 1 (5) 0

2.08 有甲, 乙, 丙三人, 每人说一句话如下:

甲说: 乙在说谎。

乙说: 丙在说谎。

丙说: 甲和乙都在说谎。

分别用整型变量 a, b, c 表示甲, 乙, 丙三个人, 且变量值为 1 表示该人说的是真话, 值为 0 表示该人在说谎。

试写出能确定谁在说谎的条件 (即逻辑表达式)。

解: 甲说: 乙在说谎。这有两种可能: 甲说的是真话, 而乙确实在说谎, 即

$$a==1 \ \&\& \ b==0 \quad \text{等价于} \quad a \ \&\& \ !b$$

或者是甲在说谎, 而乙说的是真话, 即

$$a==0 \ \&\& \ b==1 \quad \text{等价于} \quad !a \ \&\& \ b$$

由此可以得到表达式

$$a \ \&\& \ !b \ \parallel \ !a \ \&\& \ b$$

乙说: 丙在说谎。这有两种可能: 乙说的是真话, 而丙确实在说谎, 即

$$b==1 \ \&\& \ c==0 \quad \text{等价于} \quad b \ \&\& \ !c$$

或者是甲乙在说谎, 而丙说的是真话, 即

$$b==0 \ \&\& \ c==1 \quad \text{等价于} \quad !b \ \&\& \ c$$

由此可以得到表达式

$$b \ \&\& \ !c \ \parallel \ !b \ \&\& \ c$$

丙说: 甲和乙都在说谎。这有两种可能: 丙说的是真话, 而甲和乙确实都在说谎, 即

$$c==1 \ \&\& \ a+b==0 \quad \text{等价于} \quad c \ \&\& \ a+b==0$$

或者丙在说谎, 而甲和乙不都在说谎 (即甲和乙中至少有一个说的是真话), 即

$$c==0 \ \&\& \ a+b!=0 \quad \text{等价于} \quad !c \ \&\& \ a+b!=0$$

由此可以得到表达式

$$c \ \&\& \ a+b==0 \ \parallel \ !c \ \&\& \ a+b!=0$$

上述三个表达式 (条件) 是“与”的关系, 最后可以得到确定谁在说谎的逻辑表达式如下:

$$(a \ \&\& \ !b \ \parallel \ !a \ \&\& \ b) \ \&\& \ (b \ \&\& \ !c \ \parallel \ !b \ \&\& \ c) \ \&\& \ (c \ \&\& \ a+b==0 \ \parallel \ !c \ \&\& \ a+b!=0)$$

穷举每个人说真话或说谎话的各种情况, 代入上述表达式进行计算, 使上述表达式的值为 1 (真) 的情况就是正确的结果。

2.09 设有下列 C 程序:

```
#include "stdio.h"
main()
{ int a=10, b=29, c=5, d, e;
  d=(a+b)/c;   e=(a+b)%c;
  printf("d=%d, e=%d\n", d, e);
}
```

这个程序的运行结果是什么?

解: 因为整型变量 a , b , c 的值分别为 $a=10$, $b=29$, $c=5$ 。因此, 赋给整型变量 d 的值为

$$(a+b)/c=(10+29)/5=7 \text{ (注意整型运算)}$$

赋给整型变量 e 的值为

$$(a+b)\%c=(10+29)\%5=4 \text{ (注意整型运算)}$$

因此, 本程序的运行结果为

$$d=7, e=4$$

2.10 设有下列 C 程序:

```
#include "stdio.h"
main()
{ int x=20, z;
  z=++x;   z+=x;
  printf("Z1=%d\n", z);
  z=x--;   z+=x;
  printf("Z2=%d\n", z);
}
```

这个程序的运行结果是什么?

解: x 的初值为 20。在赋值语句

$$z=++x;$$

中, 变量 x 的值先加 1 后再赋给变量 z , 因此, 执行该赋值语句后, x 的值由 20 变为 21, z 的值也为 21。

赋值语句

$$z+=x;$$

等价于

$$z=z+x;$$

执行该赋值语句后, z 的值变为 $21+21=42$ 。

因此, 程序中第一个输出语句输出的结果为

$$Z1=42$$

在赋值语句

$$z=x--;$$

中, 变量 x 的值先赋给变量 z 后, x 的值再减 1, 因此, 执行该赋值语句后, z 的值变为 21, x 的值由 21 变为 20。

赋值语句

```
z+=x;
```

等价于

```
z=z+x;
```

执行该赋值语句后, z 的值变为 $21+20=41$ 。

因此, 程序中第二个输出语句输出的结果为

```
Z2=41
```

综上所述, 本程序的运行结果为

```
Z1=42
```

```
Z2=41
```

2.11 设有下列 C 程序:

```
#include "stdio.h"
main()
{ int a, b;
  float f;
  scanf("%3d%4d", &a, &b);
  f=a/b;
  printf("f=%5.2fn", f);
}
```

在运行上述程序时, 如果从键盘输入

```
2345678901↵
```

则输出结果是什么?

解: 在本程序的输入语句中, 取输入的 3 位数赋给整型变量 a , 即 $a=234$; 再顺序取 4 位数赋给整型变量 b , 即 $b=5678$ 。执行赋值语句

```
f=a/b;
```

后, 表达式 a/b 的值为 $234/5678=0$ (注意: 整型数据相除, 其结果为整型)。由于 f 是实型变量, 最后赋给变量 f 的值为实型数 0.0。

因此, 本程序的输出结果为

```
f=0.00
```

2.12 设有下列 C 程序:

```
#include "stdio.h"
main()
{ char a, b;
  scanf("%3c%4c", &a, &b);
  printf("a=%c, b=%c\n", a, b);
}
```

```

printf("a=%d, b=%d\n", a, b);
++a; --b;
printf("C1=%c, C2=%c\n", a, b);
printf("C1=%d, C2=%d\n", a, b);
}

```

在运行上述程序时，如果从键盘输入

ABCDEF GHJ

则输出结果是什么？

解：在本程序执行时，首先根据输入语句

```
scanf("%3c%4c", &a, &b);
```

取键盘输入字符串中的前三个字符 ABC，并将其中的第一个字符 A 赋给字符型变量 a；再顺序往后取四个字符 DEFG，并将其中的第一个字符 D 赋给字符型变量 b。

然后，程序中第一个输出语句的功能是以字符形式（格式说明符为 %c）输出字符型变量 a 与 b 的值，因此输出结果为

```
a=A, b=D
```

第二个输出语句的功能是以整数形式（格式说明符为 %d）输出字符型变量 a 与 b 的值，即输出它们的 ASCII 码值。而大写英文字母 A 的 ASCII 码值为 65，大写英文字母 D 的 ASCII 码值为 68。因此输出结果为

```
a=65, b=68
```

接着是依次执行以下两个表达式：

```
++a; --b;
```

其中第一个表达式是将字符型变量 a 的值加 1（即将其 ASCII 码值加 1），执行后，字符型变量 a 的 ASCII 码值由 65（字符 A）变为 66（字符 B）。第二个表达式是将字符型变量 b 的值减 1（即将其 ASCII 码值减 1），执行后，字符型变量 b 的 ASCII 码值由 68（字符 D）变为 67（字符 C）。

最后，程序中第三个输出语句的功能是以字符形式（格式说明符为 %c）输出字符型变量 a 与 b 的值，因此输出结果为

```
C1=B, C2=C
```

第四个输出语句的功能是以整数形式（格式说明符为 %d）输出字符型变量 a 与 b 的值，即输出它们的 ASCII 码值，因此输出结果为

```
C1=66, C2=67
```

综上所述，本程序的输出结果为

```
a=A, b=D
```

```
a=65, b=68
```

```
C1=B, C2=C
```

```
C1=66, C2=67
```

2.13 设有 C 程序如下：

```
#define PQ 4.5
```

```
#define ABC(x) PQ+(x*x)
```

```

#include "stdio.h"
main()
{ int a=3, b, c, d;
  b=ABC(a);
  c=ABC(a+1);
  d=2.0*ABC(a);
  printf("b=%d, c=%d, d=%d\n", b, c, d);
}

```

上述程序的运行结果是什么?

解: 在主函数前定义了一个带参的宏:

```

#define PQ 4.5
#define ABC(x) PQ+(x*x)

```

程序中第一个赋值语句

```
b=ABC(a);
```

将宏展开后为

```
b=4.5+(a*a);
```

第二个赋值语句

```
c=ABC(a+1);
```

将宏展开后为

```
c=4.5+(a+1*a+1);
```

第三个赋值语句

```
d=2.0*ABC(a);
```

将宏展开后为

```
d=2.0*4.5+(a*a);
```

将 a 的初值 3 代入计算, 可以计算出赋给变量 b , c , d (还应注意, 变量 b , c , d 均为整型) 的值分别为 13, 11, 18。

因此, 本程序的运行结果为

```
b=13, c=11, d=18
```

2.14 若将 2.13 题程序中的宏定义命令

```
#define ABC(x) PQ+(x*x)
```

改为

```
#define ABC(x) PQ+(x)*(x)
```

则该程序的运行结果是什么?

解: 在主函数前定义了一个带参的宏:

```
#define PQ 4.5
```

```
#define ABC(x) PQ+(x)*(x)
```

程序中第一个赋值语句

```
b=ABC(a);
```


将宏展开后为

```
b=4.5+(a)*(a);
```

第二个赋值语句

```
c=ABC(a+1);
```

将宏展开后为

```
c=4.5+(a+1)*(a+1);
```

第三个赋值语句

```
d=2.0*ABC(a);
```

将宏展开后为

```
d=2.0*4.5+(a)*(a);
```

将 a 的初值 3 代入计算, 可以计算出赋给变量 b , c , d (还应注意, 变量 b , c , d 均为整型) 的值分别为 13, 20, 18。

因此, 本程序的运行结果为

```
b=13, c=20, d=18
```

2.15 若将 2.13 题程序中的宏定义命令

```
#define ABC(x) PQ+(x*x)
```

改为

```
#define ABC(x) (PQ+(x)*(x))
```

则该程序的运行结果是什么?

解: 在主函数前定义了一个带参的宏:

```
#define PQ 4.5
```

```
#define ABC(x) (PQ+(x)*(x))
```

程序中第一个赋值语句

```
b=ABC(a);
```

将宏展开后为

```
b=(4.5+(a)*(a));
```

第二个赋值语句

```
c=ABC(a+1);
```

将宏展开后为

```
c=(4.5+(a+1)*(a+1));
```

第三个赋值语句

```
d=2.0*ABC(a);
```

将宏展开后为

```
d=2.0*(4.5+(a)*(a));
```

将 a 的初值 3 代入计算, 可以计算出赋给变量 b , c , d (还应注意, 变量 b , c , d 均为整型) 的值分别为 13, 20, 27。

因此, 本程序的运行结果为

```
b=13, c=20, d=27
```